

---

**Faculdade de Tecnologia de Americana “Ministro Ralph Biasi”**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**Arthur Minoru Maezono**  
**João Pedro Meneguesso**

**Gymtrace**  
**Sistema de gerenciamento para Academia Zenshin**

**Americana, SP**  
**2025**

**Arthur Minoru Maezono**  
**João Pedro Meneguesso**

**Gymtrace**  
**Sistema de gerenciamento para Academia Zenshin**

Trabalho de Conclusão de Curso desenvolvido  
em cumprimento à exigência curricular do  
Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas na área de  
concentração em Engenharia de Software.

**Orientador: Prof. Me. Clerivaldo José Roccia**

Este trabalho corresponde à versão final do  
Trabalho de Conclusão de Curso  
apresentado por Arthur Minoru Maezono e  
João Pedro Meneguesso e orientado pelo  
Prof. Me. Clerivaldo José Roccia.

**Americana, SP**  
**2025**

**FICHA CATALOGRÁFICA – Biblioteca Fatec Americana Ministro Ralph Biasi-  
CEETEPS Dados Internacionais de Catalogação-na-fonte**

MAEZONO, Arthur Minoru

Gymtrace: Sistema de gerenciamento para Academia Zenshin. / Arthur Minoru Maezono, João Pedro Meneguesso – Americana, 2025.

66f.

Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana Ministro Ralph Biasi – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Ms. Clerivaldo José Roccia

1. Administração de empresas 2. Engenharia de software 3. Sistemas de informação. I. MAEZONO, Arthur Minoru, II. MENEGUESO, João Pedro III. ROCCIA, Clerivaldo José IV. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana Ministro Ralph Biasi

CDU: 658  
681.3.05  
681518

Elaborada pelo autor por meio de sistema automático gerador de ficha catalográfica da Fatec de Americana Ministro Ralph Biasi.

Arthur Minoru Maezono  
João Pedro Meneguesso

**Gymtrace: Sistema de gerenciamento para Academia Zenshin**

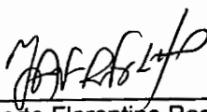
Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana Ministro Ralph Biasi.

Área de concentração: Análise e Desenvolvimento de Sistemas.

Americana, 1 de dezembro de 2025.

**Banca Examinadora:**

  
\_\_\_\_\_  
Clerivaldo José Roccia  
Mestre  
Fatec Americana "Ministro Ralph Biasi"

  
\_\_\_\_\_  
José Alberto Florentino Rodrigues Filho  
Doutor  
Fatec Americana "Ministro Ralph Biasi"

  
\_\_\_\_\_  
Fernando Mirandola  
Especialista  
Fatec Americana "Ministro Ralph Biasi"

## **AGRADECIMENTOS**

Primeiramente, agradecemos a Deus, fonte de força, sabedoria e direção em todos os momentos da nossa jornada. Foi Ele quem nos sustentou diante dos desafios e nos inspirou a perseverar até a conclusão deste trabalho.

Agradecemos também às nossas famílias, pelo amor incondicional, apoio constante, paciência e compreensão ao longo de todo o período de estudo. Cada incentivo, cada palavra de motivação e cada gesto de carinho foram fundamentais para que chegássemos até aqui.

Estendemos nossos agradecimentos aos professores da FATEC, que contribuíram para nossa formação acadêmica e profissional, compartilhando seus conhecimentos, experiências e dedicação ao ensino. Cada disciplina, cada orientação e cada desafio proposto fizeram parte essencial do nosso crescimento.

Por fim, agradecemos à própria FATEC pela oportunidade de estudar em uma instituição pública de qualidade, que nos proporcionou um ambiente de aprendizado, desenvolvimento e amadurecimento. Este trabalho é resultado não apenas do nosso esforço, mas também de tudo que recebemos ao longo de nossa trajetória acadêmica nesta instituição.

A todos que contribuíram direta ou indiretamente, nosso sincero muito obrigado.

## RESUMO

Este trabalho apresenta o desenvolvimento do sistema Gymtrace, um programa de gerenciamento feito sob medida para academias de pequeno porte no interior paulista. A pesquisa mostra que, embora o setor fitness brasileiro esteja em forte crescimento, os estabelecimentos menores frequentemente enfrentam sérios problemas na administração diária. Essas academias ainda dependem de métodos manuais, como cadernos e planilhas. Isso aumenta muito a chance de erros e causa alta inadimplência, que pode chegar a 20% ou 30% dos clientes. Além disso, os programas de gestão que existem, como os ERPs fitness, são caros (custos acima de R\$500) e complexos, o que os torna inviáveis para negócios de bairro. Diante disso, o trabalho propõe a criação e a documentação de uma solução mais simples e barata. O objetivo principal é entregar um Produto Mínimo Viável (MVP) do Gymtrace. Este MVP inclui as funções essenciais, como cadastrar alunos, controlar planos e pagamentos, registrar a frequência e gerar relatórios gerenciais básicos. A metodologia de desenvolvimento seguiu os princípios rigorosos da Engenharia de Software e usou a metodologia Ágil (Scrum), que permite fazer ajustes rápidos nos requisitos. O sistema foi construído usando o framework Laravel e o banco de dados relacional PostgreSQL. O processo incluiu o levantamento detalhado de requisitos funcionais e não funcionais e a modelagem do sistema, com a criação do Diagrama de Classes, Diagrama de Entidade e Relacionamento (DER) e Dicionário de Dados. Em conclusão, o trabalho entrega a especificação técnica completa para um sistema modular e de código aberto (open-source). Essa solução tem o objetivo de modernizar a gestão das academias locais, ajudando a reduzir erros operacionais e trazendo maior controle financeiro e previsibilidade.

**Palavras Chave:** Sistema de gerenciamento; Academias de bairro; Engenharia de Software.

## ABSTRACT

*The present text conceptualizes the development of the Gymtrace system, a management software tailored for the needs of small gyms located in the São Paulo countryside. The research identifies that the Brazilian fitness sector shows continuous expansion, but smaller establishments often face significant administrative challenges. These gyms still rely on manual methods (spreadsheets and notebooks), which greatly increases the probability of organizational errors and results in high delinquency, potentially reaching 20% to 30% of monthly members. Existing commercial software, often designed for larger chains (with monthly costs above R\$500), is considered expensive and complex for these neighborhood businesses. Given this problem, the work proposes the creation and documentation of a simpler, low-cost solution. The main goal is to deliver a Minimum Viable Product (MVP) of Gymtrace. This MVP includes essential functions like student registration, plan and payment control, attendance tracking, and generating basic managerial reports. The development methodology followed rigorous Software Engineering principles and used the Agile methodology (Scrum), which allows for quick adjustments to requirements. The system was built using the Laravel framework and the open-source relational database PostgreSQL. The process included detailed requirements gathering and structural modeling, featuring Class Diagrams, Entity-Relationship Diagrams (DER), and the Data Dictionary. In conclusion, the work delivers the complete technical specification for a modular, open-source alternative. This solution aims to modernize the administrative management of local gyms, helping to reduce operational errors and providing greater financial control and predictability.*

**Keywords:** Management system; Neighborhood gyms; Software Engineering.

## LISTA DE FIGURAS

Figura 1 - Exemplo de diagrama de caso de uso.....	8
Figura 2 - Diagrama de caso de uso do sistema.....	16
Figura 3 - Diagrama de classes do sistema.....	19
Figura 4 - Diagrama de entidade e relacionamento do sistema.....	20
Figura 5 - Tela de Login.....	28
Figura 6 - Dashboard do administrador.....	29
Figura 7 - Listagem de alunos.....	29
Figura 8 - Tela de cadastro de alunos.....	30
Figura 9 -Listagem de alunos após a criação de um novo aluno.....	31
Figura 10 – Tela de edição de alunos.....	32
Figura 11 – Listagem de alunos após atualização dos dados.....	32
Figura 12 – Tela de Listagem de Planos.....	33
Figura 13 – Tela de Cadastro de Planos.....	34
Figura 14 – Tela de Cadastro de planos após inserção do novo plano.....	35
Figura 15 - Tela de Edição de Planos (Plano Anual).....	35
Figura 16 – Tela de Edição de Planos: Dados Atualizados.....	36
Figura 17 – Tela de Listagem de Pagamentos.....	37
Figura 18 – Tela de Cadastro/Confirmação de Pagamento.....	38
Figura 19 – Tela de Edição de Pagamentos.....	38
Figura 20 – Tela de Listagem de Usuários.....	39
Figura 21 – Tela de Cadastro de Usuários.....	40
Figura 22 – Tela de Listagem de Usuários: Novo Usuário Cadastrado.....	41
Figura 23 – Tela de Edição de Usuários.....	42
Figura 24 – Tela de Edição de Usuários: Perfil Atualizado.....	42
Figura 25 – Listagem de Notificações.....	43
Figura 26 – Tela de Criação de Notificação.....	44
Figura 27 – Tela de dashboard do instrutor.....	45
Figura 28 – Listagem de Alunos para Instrutor.....	46
Figura 29 – Listagem de Presenças.....	47
Figura 30 – Tela de Registro de Presença.....	47
Figura 31 - Acesso ao repositório do projeto.....	51

## LISTA DE QUADROS

Quadro 1 - Comparativo de funcionalidades entre os concorrentes.....	11
Quadro 2 - Requisitos funcionais do sistema (contínua).....	14

Quadro 3 - Requisitos funcionais do sistema (conclusão).....	15
Quadro 4 - Requisitos não funcionais do sistema.....	15
Quadro 5 - Documentação dos casos de uso (contínua).....	17
Quadro 6 - Documentação dos casos de uso (conclusão).....	18
Quadro 7 - Dicionário de dados da entidade Usuários.....	21
Quadro 8 - Dicionário de dados da entidade Planos (contínua).....	21
Quadro 9 - Dicionário de dados da entidade Planos (conclusão).....	22
Quadro 10 - Dicionário de dados da entidade Alunos (contínua).....	22
Quadro 11 - Dicionário de dados da entidade Alunos (conclusão).....	23
Quadro 12 - Dicionário de dados da entidade Pagamentos (contínua).....	23
Quadro 14 - Dicionário de dados da entidade Pagamentos (conclusão).....	24
Quadro 15 - Dicionário de dados da entidade Presenças.....	24
Quadro 16 - Dicionário de dados da entidade Notificações.....	25



## SUMÁRIO

LISTA DE FIGURAS.....	17
LISTA DE QUADROS.....	17
1 INTRODUÇÃO.....	1
2 FUNDAMENTAÇÃO TEÓRICA.....	4
2.1 Sistemas de Informação e Gestão.....	4
2.2 Gestão em Academias.....	5
2.3 Engenharia de Software e Modelagem.....	6
2.4 Banco de Dados Relacionais E PostgreSQL.....	8
2.5 Framework Laravel.....	9
2.6 Trabalhos Relacionados.....	11
3 ENGENHARIA DE SOFTWARE.....	13
3.1 Levantamento de requisitos funcionais e não funcionais.....	13
3.2 Casos de uso.....	15
3.3 Documentação dos casos de uso.....	17
3.4 Diagrama de Classes.....	19
3.5 Diagrama de entidade e relacionamento.....	19
3.6 Dicionário de dados.....	20
4 DESENVOLVIMENTO DO PROJETO.....	26
4.1 Tecnologias Utilizadas.....	26
4.1.1 PHP 8.4.....	26
4.1.2 Laravel 12.....	27
4.1.3 PostgreSQL.....	27
4.1.4 MaterializeCSS.....	27
4.2 Desenvolvimento das Funcionalidades.....	27
4.2.1 Tela de Login.....	27
4.2.2 Módulo do Administrador.....	28
4.2.2.1 Dashboard.....	28
4.2.3 Módulo de Alunos.....	29
4.2.3.1 Listagem de Alunos.....	29
4.2.3.2 Cadastro de Alunos.....	30
4.2.3.3 Edição de Alunos.....	31
4.2.3.4 Exclusão de Alunos.....	33
4.2.4 Módulo de Planos.....	33
4.2.4.1 Listagem de Planos.....	33
4.2.4.2 Cadastro de Planos.....	34
4.2.4.3 Edição de Planos.....	35
4.2.4.4 Exclusão de Planos.....	36
4.2.5 Módulo de Pagamentos.....	36
4.2.5.1 Listagem de Pagamentos.....	37

4.2.5.2 Registro de Pagamentos.....	37
4.2.5.3 Edição de Pagamentos.....	38
4.2.5.4 Exclusão de Pagamentos.....	39
4.2.6 Módulo de Usuários.....	39
4.2.6.1 Listagem de Usuários.....	39
4.2.6.2 Cadastro de Usuários.....	40
4.2.6.3 Edição de Usuários.....	41
4.2.6.4 Exclusão de Usuários.....	43
4.2.7 Módulo de Notificações.....	43
4.2.7.1 Listagem de Notificações.....	43
4.2.7.1 Criação de Notificações.....	43
4.3 Módulos destinados ao Instrutor.....	44
4.3.1 Dashboard (Instrutor).....	45
4.3.2 Consulta de Alunos (Instrutor).....	45
4.3.3 Listagem de Alunos (Somente consulta).....	45
4.3.4 Registro de Presenças.....	46
4.3.5 Listagem de Presenças.....	46
4.3.6 Registro de Nova Presença.....	47
4.3.7 Limitações de Acesso do Instrutor.....	48
4.4 Considerações sobre Segurança.....	48
5 CONSIDERAÇÕES FINAIS.....	49
5.1 Resultados Alcançados.....	49
5.2 Impacto esperado nas Academias.....	49
5.3 Limitações do Estudo.....	50
5.4 Propostas de desenvolvimento e melhorias.....	50
5.5 Acesso aos repositórios.....	51
REFERÊNCIAS BIBLIOGRÁFICAS.....	52

## 1 INTRODUÇÃO

O setor fitness no Brasil tem demonstrado uma expansão contínua, resultando na proliferação de academias em todo o território nacional. Segundo dados da Associação Brasileira de Academias (ACAD Brasil, 2024), o país conta com quase 57 mil academias, um faturamento anual superior a R\$17 bilhões e cerca de 15 milhões de matriculados, posicionando-se como o maior mercado global nesse segmento. Contudo, estabelecimentos de menor porte, como as academias de bairro localizadas em regiões como o interior paulista, frequentemente enfrentam obstáculos na administração de suas rotinas, incluindo a gestão de inscrições, mensalidades, planos e o monitoramento da frequência dos clientes. Esses estabelecimentos atendem predominantemente a um público específico da localidade, como trabalhadores autônomos e famílias de baixa a média renda, que demandam soluções simplificadas e economicamente acessíveis, considerando limitações como conectividade instável à internet e infraestrutura básica.

Muitas dessas academias ainda dependem de métodos manuais, como planilhas em Excel e anotações em cadernos, para controlar suas operações. Essa abordagem eleva a probabilidade de erros organizacionais específicos, como duplicação de cadastros de alunos, perda de registros de frequência ou inconsistências em pagamentos, o que pode resultar em inadimplência de até 20-30% dos mensalistas — um impacto financeiro significativo para negócios com margens apertadas, podendo levar a perdas anuais de milhares de reais e até o fechamento de unidades. Além disso, dificulta a análise do crescimento do negócio, impedindo projeções precisas de receita ou identificação de padrões de evasão de alunos. A carência de sistemas informatizados acessíveis e adequados à realidade das pequenas empresas — especialmente em contextos regionais com restrições de infraestrutura — torna a gestão mais custosa e menos eficaz.

Neste cenário, os sistemas de informação, definidos como conjuntos integrados de hardware, software, dados e procedimentos que coletam, processam e distribuem informações para apoiar a tomada de decisões (Laudon e Laudon, 2014), surgem como soluções estratégicas para otimizar a gestão. A tecnologia permite que as empresas integrem a coleta, o armazenamento e o processamento de dados, aprimorando a eficiência operacional e reduzindo custos. Todavia, grande parte dos softwares disponíveis, como plataformas como Gympass ou ERPs fitness

especializados (ex.: SmartFit Manager), é projetada para redes de médio e grande porte, com custos mensais acima de R\$500 e complexidade que exigem treinamento especializado. Essas soluções frequentemente ignoram lacunas críticas para pequenas academias, como suporte a operações offline, interfaces intuitivas para não-técnicos e relatórios personalizados para inadimplência local, tornando-as inviáveis para empreendimentos de bairro.

Frente a essa problemática, o presente trabalho propõe o desenvolvimento e a documentação de um sistema de gerenciamento para academias de pequeno porte no interior paulista, utilizando a tecnologia Laravel e o banco de dados PostgreSQL. O objetivo é entregar um Produto Mínimo Viável (MVP) — uma versão inicial e funcional de um software com as funcionalidades essenciais para validar a ideia e iterar com base em feedback real (Ries, 2011) —, incluindo o cadastro de alunos, controle de planos e pagamentos, registro de frequência e a geração de relatórios gerenciais básicos, como resumos de inadimplência e retenção de clientes.

Além da implementação técnica, esta pesquisa visa analisar as deficiências das soluções concorrentes para oferecer uma alternativa simplificada, de baixo custo (desenvolvimento open-source e hospedagem acessível) e alinhada às necessidades de academias locais, promovendo maior retenção de alunos por meio de uma experiência mais fluida. Com isso, espera-se colaborar para a modernização administrativa desses negócios, otimizando recursos, ganhando eficiência e melhorando a satisfação dos usuários.

Para delimitar o escopo, este trabalho não abordará aspectos como integração com ferramentas de marketing digital, controle detalhado de treinos personalizados ou módulos de e-commerce para vendas de suplementos, focando exclusivamente na gestão administrativa básica. Da mesma forma, o sistema não contemplará integrações com dispositivos IoT para monitoramento biométrico, priorizando acessibilidade e simplicidade para ambientes com baixa conectividade.

Este documento está estruturado da seguinte maneira: o segundo capítulo aborda o referencial teórico sobre sistemas de informação, engenharia de software e tecnologias de desenvolvimento, além de analisar produtos similares. O terceiro capítulo detalha a metodologia, incluindo o tipo de pesquisa e as ferramentas utilizadas. O quarto capítulo descreve o processo de desenvolvimento do sistema,

dos requisitos aos testes. Finalmente, o quinto capítulo apresenta as conclusões, os resultados alcançados e as sugestões para pesquisas futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos que embasam o tema do trabalho. Inicialmente, aborda-se a Engenharia de Software, disciplina responsável pela criação, desenvolvimento e manutenção de programas de computador mediante métodos sistemáticos e planejados. Essa disciplina engloba etapas como planejamento, projeto, implementação, testes e manutenção, as quais garantem a qualidade do software e sua aderência aos requisitos dos usuários. Diversas abordagens podem ser adotadas, desde modelos tradicionais, como o cascata, até metodologias ágeis, que priorizam flexibilidade e ciclos de desenvolvimento curtos.

Também é discutida a gestão de academias, que é o conjunto de ações para administrar bem esses lugares, cuidando das pessoas que trabalham, dos clientes, dos equipamentos e das rotinas diárias. A utilização de sistemas digitais é fundamental para otimizar a gestão, aumentando a eficiência operacional e melhorando a experiência do cliente.

O capítulo também expõe os principais desafios e métodos empregados no desenvolvimento de sistemas de gestão para academias, destacando a necessidade de acompanhamento contínuo das mudanças mercadológicas e tecnológicas. Esse embasamento teórico possibilita uma compreensão mais aprofundada das escolhas metodológicas e tecnológicas que fundamentam o desenvolvimento do projeto, detalhado nos capítulos subsequentes.

### 2.1 Sistemas de Informação e Gestão

Os Sistemas de Informação (SI) representam muito mais do que meros conjuntos de software e hardware; eles constituem um fundamento estratégico para a operação e gestão modernas. De forma integrada, combinam pessoas, processos, dados e tecnologia com o objetivo primordial de coletar, processar, armazenar e disseminar informações para suportar a tomada de decisão, a coordenação e o controle dentro das organizações. Conforme seminalmente definido por Laudon e Laudon (2018, p. 47), um sistema de informação é "um conjunto de componentes inter-relacionados que trabalham para coletar, processar, armazenar e disseminar informações para apoiar a tomada de decisões, a coordenação, o controle, a análise e a visualização em uma organização". Eles são indispensáveis para que empresas

de diferentes portes otimizem operações, obtenham eficiência e, crucialmente, mantenham vantagem competitiva em um mercado globalizado.

No ecossistema das Micro e Pequenas Empresas (MPEs), a adoção de Sistemas de Informação informatizados representa frequentemente um marco decisivo. Stair e Reynolds (2015) argumentam que a tecnologia da informação atua como um grande equalizador, permitindo que pequenos negócios compitam com organizações maiores. Eles afirmam que tais sistemas "contribuem para padronizar processos, reduzir erros manuais e aumentar a confiabilidade dos dados disponíveis para gestores". No contexto específico das academias de bairro, essa solução tecnológica se materializa na centralização do controle de matrículas, pagamentos, frequência e planos de treino em uma única plataforma. Essa integração não apenas reduz custos administrativos e minimiza retrabalho, mas também eleva substancialmente a qualidade do atendimento ao cliente, permitindo um serviço mais personalizado e ágil.

Dados recentes da Associação Brasileira de Academias (ACAD Brasil, 2025) indicam que o país conta com mais de 56 mil academias em funcionamento, movimentando um faturamento anual superior a R\$8 bilhões e empregando diretamente cerca de 400 mil profissionais. Essas estatísticas reforçam a importância de sistemas informatizados para otimizar a gestão, especialmente nos negócios de pequeno porte, que compõem a grande maioria do setor em âmbito local.

## 2.2 Gestão em Academias

O setor fitness brasileiro é um dos mais dinâmicos e promissores do mundo. Dados da Associação Brasileira de Academias (ACAD BRASIL, 2023) consolidam essa posição, indicando que o país conta com uma rede vasta de estabelecimentos e milhões de frequentadores ativos, um mercado que movimenta bilhões de reais anualmente. No entanto, por trás desse cenário de expansão contínua, as academias de pequeno porte enfrentam desafios estruturais críticos. Problemas como a alta taxa de evasão de alunos (churn), a inadimplência – que frequentemente supera a média de outros setores de serviços – e a carência de ferramentas tecnológicas acessíveis e de fácil implementação representam graves ameaças à sua sustentabilidade financeira e operacional.

Nesse contexto desafiador, a implantação de um sistema informatizado de gestão surge não como um luxo, mas como uma necessidade estratégica. Tal sistema pode oferecer benefícios diretos e mensuráveis, como um controle financeiro mais rigoroso – com acompanhamento de fluxo de caixa, receitas e despesas –, um acompanhamento mais preciso da evolução física e da adesão dos alunos aos treinos, e o estabelecimento de canais de comunicação mais eficientes entre gestores, instrutores e clientes. Funcionalidades aparentemente simples, como notificações automáticas por e-mail ou WhatsApp sobre o vencimento de mensalidades e a geração de relatórios financeiros básicos, podem ter um impacto transformador para academias locais que operam com recursos limitados, conferindo maior previsibilidade e rigor administrativo à gestão.

Estudos de caso reais confirmam a eficácia dessa estratégia. Por exemplo, o sistema Next Fit, implantado em academias de pequeno porte, conseguiu reduzir a inadimplência em até 25%, adotando funcionalidades como notificações automáticas sobre vencimentos e a geração de relatórios financeiros básicos podem impactar significativamente a gestão de academias locais. Estudos de caso indicam que a automação de lembretes de pagamento reduz a inadimplência em até 25% (LOPES; DANTAS, 2024), trazendo previsibilidade financeira essencial para negócios de pequeno porte.

### **2.3 Engenharia de Software e Modelagem**

Considerando a criticidade dos sistemas de gestão para a operação de pequenos negócios, a aplicação de princípios consolidados da Engenharia de Software torna-se imperativa para garantir confiabilidade, manutenibilidade e aderência aos requisitos do domínio do problema. Essa abordagem assegura que o sistema seja confiável, escalável e alinhado às demandas reais dos usuários, minimizando riscos, facilitando manutenções futuras e garantindo qualidade técnica e operacional.

Conforme Pressman e Maxim (2016), a Engenharia de Software estabelece um conjunto de processos, métodos e ferramentas para o desenvolvimento profissional de sistemas computacionais. Embora outras metodologias ágeis como Kanban (foco no fluxo contínuo) e *Extreme Programming* (ênfase em práticas técnicas) sejam alternativas viáveis, o Scrum foi selecionado por oferecer uma

estrutura bem definida com cerimônias e artefatos específicos, facilitando o gerenciamento do ciclo de desenvolvimento em um contexto acadêmico com prazos delimitados.

A utilização de técnicas de modelagem, como a UML (Unified Modeling Language, versão 2.5 atualizada pelo Object Management Group em 2015), possibilita representar graficamente processos, atores, relacionamentos e fluxos de informação, facilitando a comunicação entre desenvolvedores, stakeholders e usuários finais. Diagramas como casos de uso (para mapear interações do usuário), classes (para definir estruturas de dados) e atividades (para fluxos de processos) contribuem para reduzir ambiguidades, mitigar interpretações equivocadas e subsidiar o processo de documentação, essencial em projetos acadêmicos e corporativos.

Como exemplo das técnicas de modelagem com UML, a Figura 1 mostra um diagrama de casos de uso para um sistema de academia, com três tipos de usuários:

- Aluno: Realiza ações como fazer check-in, ver horários de aula e pedir reposição;
- Instrutor: Cuida de tarefas como registrar presença e fazer avaliações físicas;
- Recepção: Responsável por funções administrativas, como matrículas e controle de planos.

Esse exemplo demonstra como a UML possibilita a compreensão, por meio de representação visual, das funcionalidades do sistema e da distribuição de responsabilidades entre os usuários.

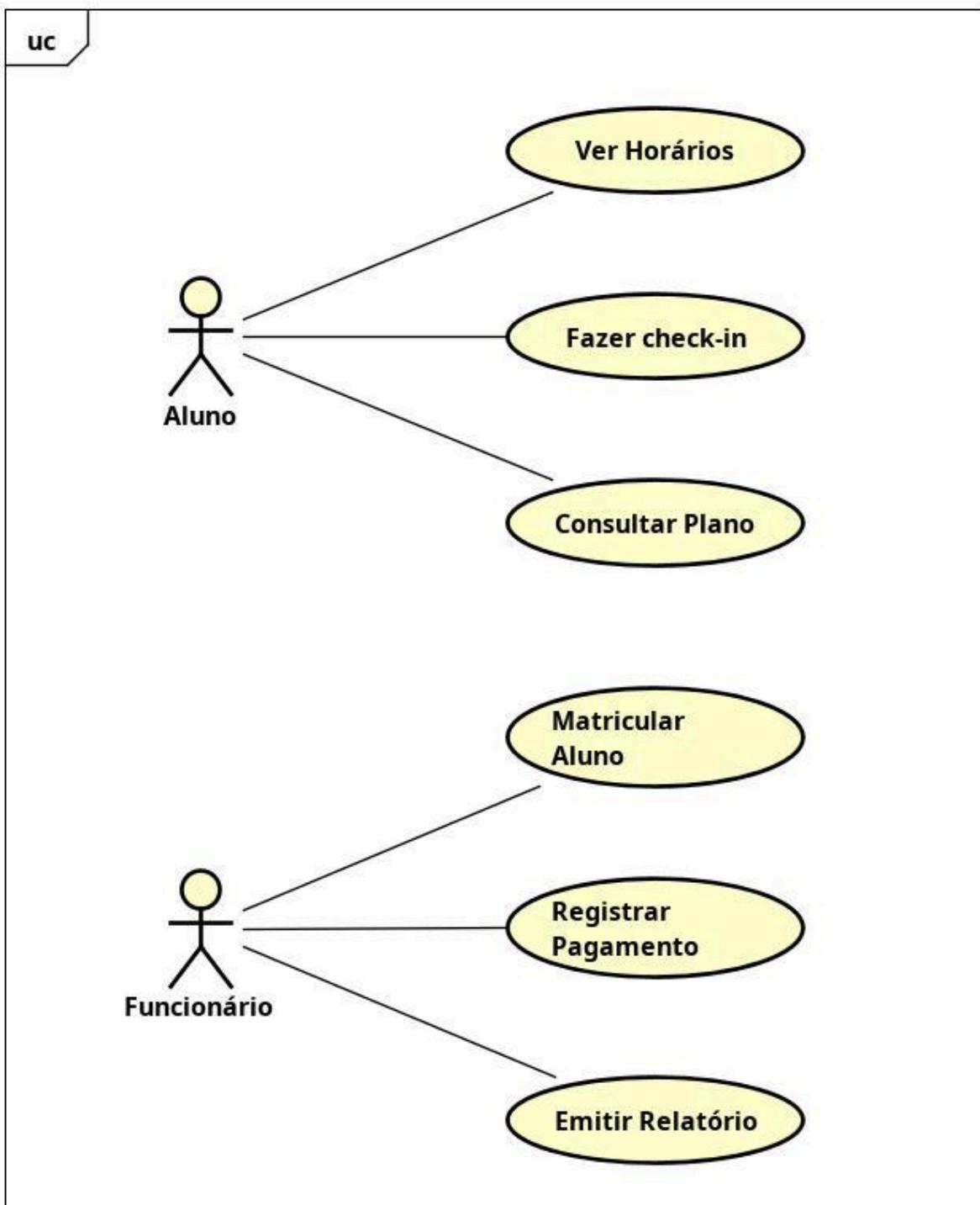


Figura 1 - Exemplo de diagrama de caso de uso

Fonte: Elaborado pelo autor (2025)

## 2.4 Banco de Dados Relacionais E PostgreSQL

O banco de dados é um componente central de qualquer sistema de informação, atuando como repositório organizado e seguro para o armazenamento e recuperação de dados. Os modelos relacionais, amplamente utilizados no mercado desde os anos 1970 (inspirados no trabalho de E.F. Codd), são baseados em tabelas que representam entidades (ex.: alunos, planos de treino) e seus relacionamentos (ex.: chaves estrangeiras para vincular pagamentos a matrículas). Conforme Date (2019), em An Introduction to Database Systems (8<sup>a</sup> edição), bancos relacionais são adequados quando há necessidade de consistência (via normalização para evitar redundâncias), integridade (regras de constraints) e manipulação estruturada dos dados por meio de linguagens como SQL, suportando consultas complexas e transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

O PostgreSQL destaca-se entre os sistemas de gerenciamento de banco de dados (SGBD) relacionais por ser gratuito, de código aberto (licença PostgreSQL) e altamente robusto, com adoção crescente em 70% dos desenvolvedores de backend segundo o Stack Overflow Survey (2024). Além da conformidade total com os padrões SQL (ANSI/ISO), oferece recursos avançados como suporte nativo a dados JSON e XML para flexibilidade em aplicações modernas, extensibilidade via linguagens como PL/pgSQL, e mecanismos de replicação (streaming replication) e backup automatizado (pg\_dump). Essas características técnicas, corroboradas por benchmarks independentes que demonstram superioridade em operações de leitura intensiva e concorrência (STACK OVERFLOW, 2024), tornam o PostgreSQL adequado para o cenário de academias em crescimento. A integração nativa com o Laravel via Eloquent ORM e a ausência de custos de licenciamento consolidam sua seleção como SGBD para este projeto. No cenário brasileiro, o PostgreSQL é amplamente usado em PMEs do setor de serviços, conforme relatório da ABES (Associação Brasileira das Empresas de Software, 2023), por sua estabilidade e fácil integração com frameworks web.

## 2.5 Framework Laravel

Laravel é um framework web baseado na linguagem PHP que adota a arquitetura MVC (Model-View-Controller), separando responsabilidades para promover código limpo e modular. Desenvolvido inicialmente em 2011 por Taylor

Otwell, tornou-se um dos mais utilizados mundialmente – ocupando o segundo lugar em popularidade para PHP no Stack Overflow Survey (2024) – devido à simplicidade na estruturação de projetos, à sintaxe elegante e à grande quantidade de recursos disponíveis nativamente, que eliminam a necessidade de bibliotecas externas para tarefas comuns.

Segundo a documentação oficial (LARAVEL, 2024), o framework inclui ferramentas poderosas como o ORM Eloquent (para mapeamento objeto-relacional, facilitando interações com bancos como PostgreSQL), migrations (para versionamento de esquemas de banco de dados de forma colaborativa), autenticação integrada (com Laravel Sanctum para APIs) e suporte a filas de processamento (via Laravel Horizon para tarefas assíncronas, como envio de e-mails). Essas funcionalidades, conforme a documentação oficial, permitem que o uso de migrations e Eloquent reduza o tempo de prototipagem em comparação com o PHP puro, minimize a ocorrência de erros por meio de validações automáticas e mantenham práticas consolidadas de programação, como injeção de dependências e testes unitários.

A escolha do Laravel foi fundamentada em critérios técnicos objetivos. Benchmarks comparativos demonstram que o framework oferece um equilíbrio ideal entre performance e produtividade do desenvolvedor (STACK OVERFLOW, 2024). Especificamente para o domínio de sistemas de gestão, características como o Eloquent ORM simplificam operações complexas de banco de dados, enquanto o sistema de *migrations* assegura a evolução controlada do esquema de dados. Adicionalmente, a robustez dos mecanismos de segurança nativos, incluindo proteção contra CSRF e hash de senhas via bcrypt, atende aos requisitos de confidencialidade inerentes a dados financeiros e pessoais.

O framework promove práticas modernas de manutenção e testes, favorecendo a criação de testes unitários e integrações contínuas, o que assegura maior estabilidade e qualidade na aplicação. Para academias de bairro, essa característica viabiliza o desenvolvimento ágil de módulos como controle de acesso e relatórios, adequados a contextos com restrições orçamentárias.

## 2.6 Trabalhos Relacionados

O mercado brasileiro já dispõe de sistemas especializados para academias, consolidando um ecossistema maduro de soluções SaaS (Software as a Service). Entre os principais sistemas, destacam-se o Next Fit, o Tecnofit e o Hubfit, que atendem a diferentes necessidades do setor fitness.

A seguir, um quadro comparativo resumido:

Quadro 1 - Comparativo de funcionalidades entre os concorrentes

Sistema	Funcionalidades Principais	Custo Mensal	Público-Alvo
Next Fit	Cadastro de alunos, controle financeiro, cobranças automáticas, app móvel	A partir de R\$150-600	Pequenas a grandes academias
Tecnofit	Gestão de estúdios e boxes, agendamento de aulas, integração com wearables	Variável, plano personalizado	Academias, estúdios e boxes de CrossFit
Hubfit	Integração com hardware (catracas, sensores), relatórios preditivos	A partir de R\$200-700	Academias médias e grandes

Fonte: Elaborado pelo autor (2025)

Apesar da robustez e escalabilidade dessas soluções, academias de bairro frequentemente enfrentam barreiras como custo elevado e complexidade de implantação, além da necessidade de treinamento e suporte técnico constante. Essa realidade é confirmada por Lopes e Dantas (2024), que em sua análise sobre a revolução digital nas academias de pequeno porte, identificaram que a resistência à mudança cultural e as limitações de recursos financeiros representam entraves significativos na adoção tecnológica, comprometendo a eficiência operacional e a personalização dos serviços.

Além deste estudo, a pesquisa de Almeida e Silva (2020) corrobora essa perspectiva ao desenvolver uma plataforma web integrada para gestão de academias, demonstrando na prática que sistemas simplificados e modulares - com funcionalidades essenciais como cadastro de praticantes, controle de treinos e acompanhamento de evolução - são viáveis técnica e operacionalmente para o

contexto das pequenas academias, atendendo à demanda por ferramentas acessíveis, intuitivas e focadas nas operações básicas.

Dessa forma, o Gymtrace diferencia-se por oferecer uma solução open-source, modular e gratuita, com foco exclusivo em academias de bairro e pequenas empresas. O sistema propõe reduzir custos, facilitar a implantação e promover uma experiência de gerenciamento acessível, preenchendo as lacunas existentes deixadas por softwares comercialmente robustos, porém onerados e complexos.

### 3 ENGENHARIA DE SOFTWARE

Este capítulo apresenta a especificação técnica do sistema Gymtrace, detalhando os elementos fundamentais que orientaram seu desenvolvimento. A abordagem seguiu os princípios da engenharia de software, com ênfase no levantamento de requisitos e na modelagem do sistema.

Inicialmente, foram definidos os requisitos funcionais e não funcionais. Conforme Sommerville (2016), "os requisitos funcionais especificam ou descrevem as funções que o sistema deve ser capaz de executar", como cadastro de alunos, gestão de planos e registro de pagamentos. Já os requisitos não funcionais, que "especificam restrições e critérios de qualidade" (SOMMERVILLE, 2016), incluíram aspectos de usabilidade, segurança e desempenho.

Em seguida, foram elaborados os casos de uso, que, segundo Pressman e Maxim (2016), "proporcionam uma descrição clara e comprehensível dos requisitos funcionais". O diagrama correspondente (Figura 2) representa as interações entre administradores, instrutores e alunos com o sistema, enquanto a documentação detalhada especifica objetivos, atores e condições para cada caso.

A modelagem estrutural foi complementada pelo diagrama de classes (Figura 3), que define a estrutura estática do sistema, e pelo diagrama de entidade e relacionamento (Figura 4), que representa visualmente as entidades e seus relacionamentos. Finalmente, o dicionário de dados (Tabelas 4 a 9) organizou de forma padronizada todos os elementos de dados do sistema, assegurando consistência na implementação.

#### 3.1 Levantamento de requisitos funcionais e não funcionais

O levantamento de requisitos constitui uma etapa fundamental no desenvolvimento de sistemas de software, sendo o processo no qual são identificadas as funcionalidades e atributos que o sistema deve incorporar para satisfazer às necessidades dos usuários.

Os requisitos funcionais definem as ações que o sistema deve realizar para cumprir seu propósito. Segundo Sommerville (2016), "os requisitos funcionais especificam ou descrevem as funções que o sistema deve ser capaz de executar". Exemplos comuns incluem operações básicas como inserir, alterar e excluir dados,

gerar relatórios, autenticar usuários, entre outros. Esses requisitos refletem diretamente o comportamento esperado pelo usuário final.

Por outro lado, os requisitos não funcionais tratam de como o sistema deve realizar essas funções, preocupando-se com atributos de qualidade que influenciam a usabilidade, desempenho, segurança e confiabilidade do software. Sommerville destaca que os requisitos não funcionais "especificam restrições e critérios de qualidade que o sistema deve atender para ser aceitável". Eles são essenciais para garantir que o sistema seja eficiente, seguro e estável, podendo abranger aspectos como tempo de resposta, disponibilidade, escalabilidade e compatibilidade.

A distinção entre requisitos funcionais e não funcionais é crucial, pois ambos impactam diretamente no projeto e na arquitetura do software. Enquanto os requisitos funcionais determinam o que o sistema fará, os requisitos não funcionais condicionam a forma como essas funções serão entregues, assegurando padrões de qualidade e satisfação do usuário. Esse levantamento detalhado serve como base para todas as fases seguintes de desenvolvimento, garantindo que o produto final esteja alinhado às expectativas e necessidades apresentadas durante a fase inicial do projeto.

Para facilitar a compreensão, a seguir encontram-se as tabelas que apresentam os requisitos funcionais e não funcionais do sistema, conforme identificados durante o levantamento do projeto.

Os Quadros 2 e 3 apresentam os requisitos funcionais levantados para o sistema.

Quadro 2 - Requisitos funcionais do sistema (contínua)

Identificação	Requisito Funcional	Prioridade
RFBD1	Sistema deve possuir um banco de dados relacional PostgreSQL	Essencial
RFBD2	Banco de dados deve possuir tabela para cadastro de usuários	Essencial
RFBD3	Banco de dados deve possuir tabela para cadastro de alunos	Essencial
RFBD4	Banco de dados deve possuir tabela para planos e mensalidades	Essencial
RFBD5	Banco de dados deve possuir tabela para controle de pagamentos	Essencial

Quadro 3 - Requisitos funcionais do sistema (conclusão)

Identificação	Requisito Funcional	Prioridade
RFA01	Sistema deve possuir API para operações CRUD	Essencial
RFA02	API deve garantir segurança e autenticação para acesso	Essencial
RFS01	Sistema deve possuir site acessível para usuários autorizados	Essencial
RFS02	Site deve permitir cadastro, alteração e consulta de dados	Essencial
RFS03	Site deve gerar relatórios gerenciais básicos	Essencial
RFS04	Site deve enviar notificações automáticas (e-mail/Whatsapp) sobre vencimentos	Desejável
RFS05	Site deve possuir área administrativa restrita	Essencial

Fonte: Elaborado pelo autor (2025)

O Quadro 4 apresenta os requisitos não funcionais levantados para o sistema.

Quadro 4 - Requisitos não funcionais do sistema

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	O site e o aplicativo devem ter design amigável	Usabilidade	Essencial
RNF002	O site e o aplicativo devem ser responsivos	Usabilidade	Essencial
RNF003	Os dados manipulados devem estar em segurança	Segurança	Essencial
RNF004	O site deve ser rápido	Desempenho	Importante
RNF005	Os dados apresentados devem condizer com a pesquisa	Confiabilidade	Essencial
RNF006	Dados sensíveis só estarão disponíveis para administrador	Segurança	Essencial
RNF007	Identidade visual deve ser padronizada no site e no app	Padrões	Importante

Fonte: Elaborado pelo autor (2025)

### 3.2 Casos de uso

Os casos de uso são uma ferramenta fundamental na engenharia de software para descrever como os usuários interagem com o sistema, ajudando a identificar as funcionalidades necessárias. Segundo Pressman e Maxim (2016), casos de uso proporcionam uma descrição clara e comprehensível dos requisitos funcionais, mostrando as ações que os usuários podem realizar no sistema.

De acordo com Sommerville (2016), eles representam uma sequência de atividades que ilustram os passos que um usuário deve seguir para alcançar um objetivo específico dentro do sistema. Cada caso de uso ajuda a definir o que o sistema deve fazer, sem entrar em detalhes técnicos da implementação.

O diagrama de casos de uso, mostrado neste capítulo, oferece uma visão geral dos atores (usuários ou sistemas externos) e dos casos de uso (funcionalidades) que compõem o sistema, facilitando a comunicação entre a equipe de desenvolvimento e os envolvidos no projeto.

Assim, os casos de uso são essenciais para planejar, documentar e validar os requisitos do sistema, garantindo que o software atenda às necessidades reais dos usuários.

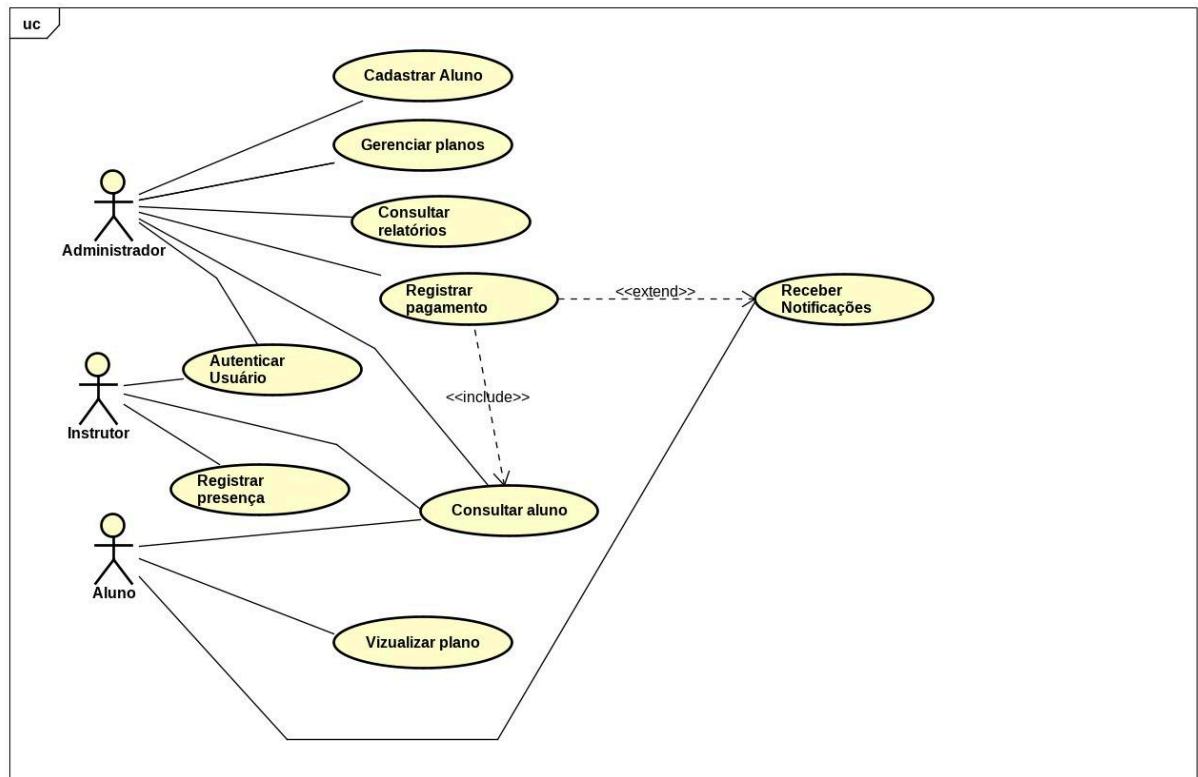


Figura 2 - Diagrama de caso de uso do sistema

Fonte: Elaborado pelo autor (2025)

### 3.3 Documentação dos casos de uso

Este capítulo apresenta a descrição resumida de cada caso de uso do sistema, destacando seus objetivos, os atores envolvidos, e as condições necessárias para seu início e término. A documentação busca facilitar o entendimento das funcionalidades e a comunicação entre as partes interessadas no projeto.

Quadro 5 - Documentação dos casos de uso (contínua)

Número do Caso de Uso	Nome do Caso de Uso	Descrição	Atores Envolvidos	Pré-condição	Pós-condição
1	Cadastrar Aluno	Permite ao administrador incluir um novo aluno no sistema, registrando suas informações.	Administrador	O administrador deve estar autenticado	Aluno cadastrado e disponível no sistema
2	Gerenciar Planos	Permite ao administrador criar, modificar ou remover planos de treino e mensalidades para alunos.	Administrador	O administrador deve estar autenticado	Plano criado, alterado ou excluído com sucesso
3	Consultar Relatórios	O administrador pode visualizar informações e estatísticas sobre alunos, planos e pagamentos.	Administrador	O administrador deve estar autenticado	Relatório consultado e disponível

Quadro 6 - Documentação dos casos de uso (conclusão)

Número do Caso de Uso	Nome do Caso de Uso	Descrição	Atores Envolvidos	Pré-condição	Pós-condição
4	Registrar Pagamento	Permite ao administrador marcar que um pagamento foi feito por um aluno, atualizando o histórico financeiro.	Administrador	O administrador deve estar autenticado	Pagamento registrado no sistema
5	Receber Notificações	O administrador é avisado automaticamente sobre eventos importantes, como pagamentos e atualizações no sistema.	Administrador	Pagamento registrado ou evento importante	Notificação recebida pelo administrador
7	Registrar Presença	O instrutor marca sua presença ou a presença dos alunos nas atividades e aulas.	Instrutor	Instrutor autenticado no sistema	Presença registrada e disponível para consulta
8	Consultar Aluno	Consultar dados pessoais, informações de matrícula, plano e presença dos alunos.	Administrador, Instrutor	Usuário autenticado	Informações do aluno exibidas para consulta
9	Visualizar Plano	O aluno acessa o sistema para ver os detalhes do seu plano de treino ou mensalidade.	Aluno	Aluno deve estar autenticado no sistema	Plano exibido para o aluno

Fonte: Elaborado pelo autor (2025)

### 3.4 Diagrama de Classes

O diagrama de classes constitui uma representação gráfica que evidencia os componentes principais de um sistema e seus respectivos relacionamentos. Cada elemento, denominado classe, é constituído por um identificador, propriedades (atributos) e operações (métodos). Este diagrama auxilia na organização e compreensão da arquitetura do sistema, apresentando de maneira estruturada os objetos envolvidos, suas responsabilidades e suas interações. Também auxilia na criação do banco de dados, pois cada classe pode virar uma tabela. Dessa forma, viabiliza-se o planejamento e a construção de sistemas com maior clareza e organização estrutural.

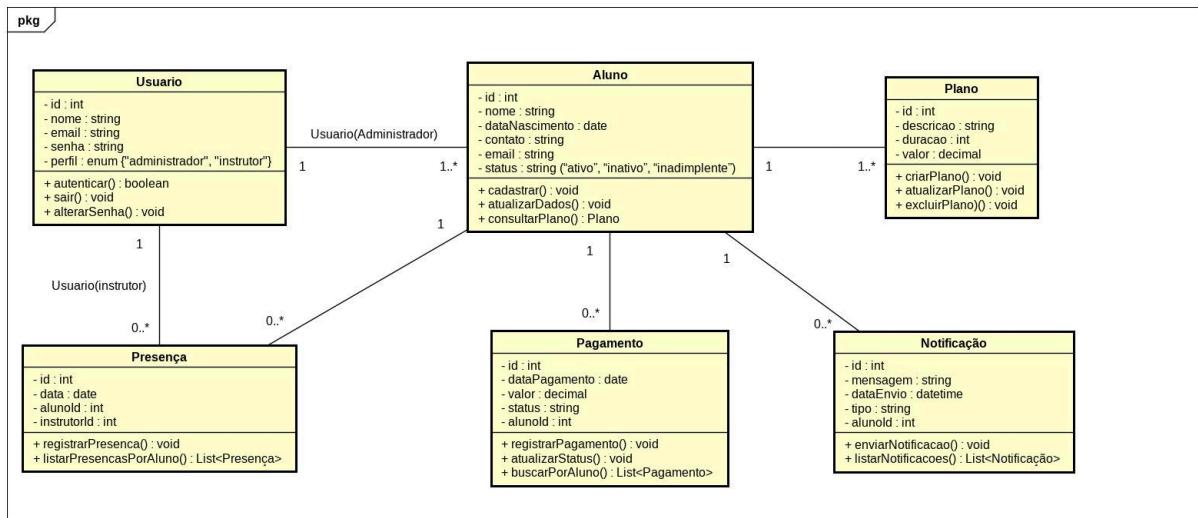


Figura 3 - Diagrama de classes do sistema

Fonte: Elaborado pelo autor (2025)

### 3.5 Diagrama de entidade e relacionamento

O diagrama de entidade e relacionamento (DER) é uma representação visual que mostra as entidades presentes em um sistema e como elas se conectam entre si. Segundo Sommerville (2016), entidades são objetos ou conceitos importantes para o domínio do sistema, e cada uma possui características próprias chamadas atributos. O DER utiliza símbolos simples, como retângulos para entidades e linhas para demonstrar as relações entre elas, indicando também a quantidade dessas relações, conhecida como cardinalidade.

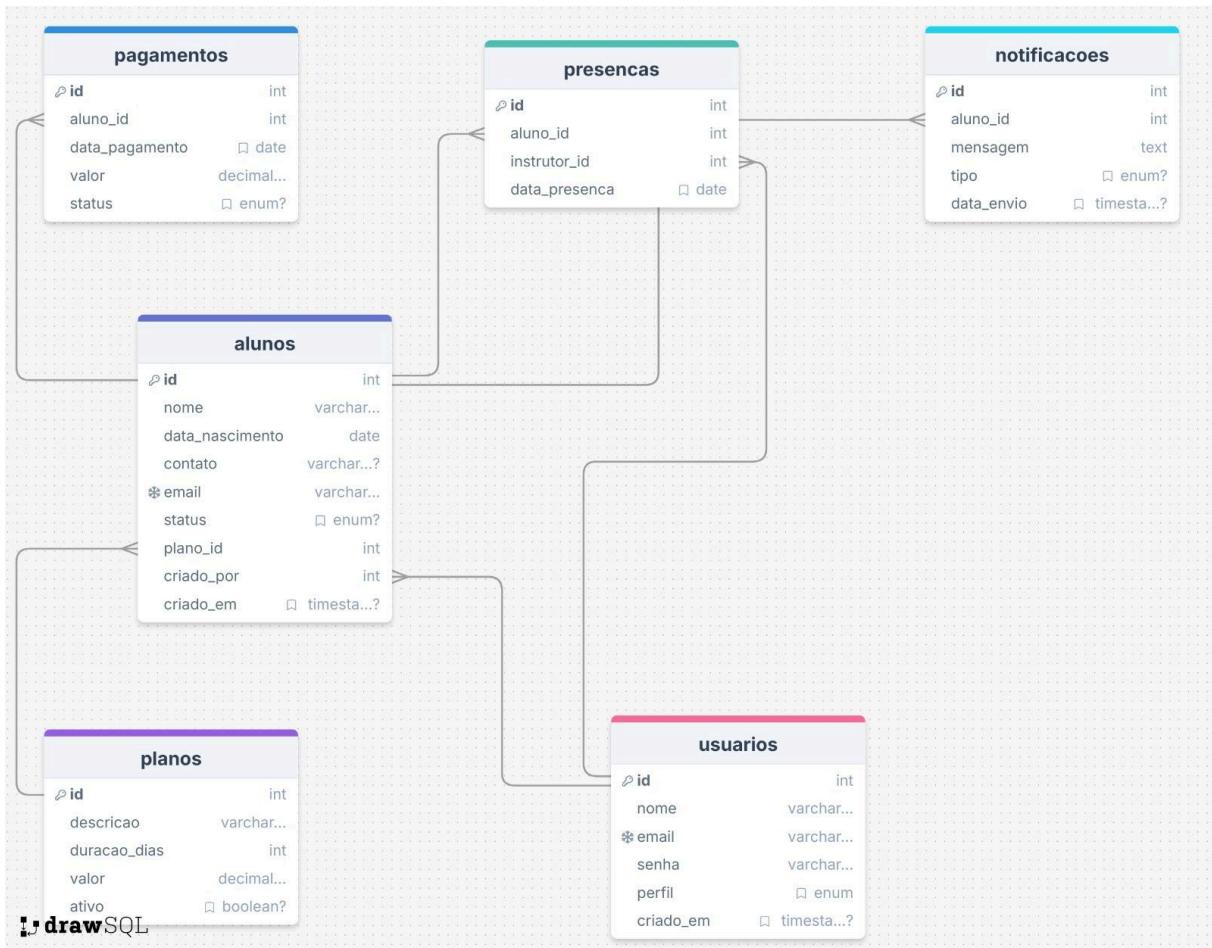


Figura 4 - Diagrama de entidade e relacionamento do sistema

Fonte: Elaborado pelo autor (2025)

### 3.6 Dicionário de dados

O Dicionário de Dados (DD) consiste em um repertório organizado de todos os elementos de dados pertinentes ao sistema. Ele inclui informações sobre cada elemento, como a entidade à qual pertence, o nome do atributo, sua classe, domínio, tamanho e uma descrição clara. As tabelas que compõem o DD devem conter esses campos para garantir que os dados sejam bem compreendidos e usados corretamente durante o desenvolvimento e manutenção do sistema.

O Quadro 8 apresenta o DD da entidade Usuários.

Quadro 7 - Dicionário de dados da entidade Usuários

<b>Entidade: Usuários</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
id	Determinante	Numérico	Inteiro 32 bits	Identificador único do usuário (chave primária).
nome	Simples	Texto	100	Nome completo do usuário do sistema.
email	Simples	Texto	100	Endereço de e-mail utilizado para login e identificação.
senha	Simples	Texto	255	Senha criptografada de acesso ao sistema.
perfil	Simples	Texto (Enum)	20	Define o tipo de usuário: "administrador" ou "instrutor".
criado_em	Simples	Data/Hora	Timestamp	Data e hora em que o registro foi criado.

Fonte: Elaborado pelo autor (2025)

Os Quadros 8 e 9 apresentam o DD da entidade Planos.

Quadro 8 - Dicionário de dados da entidade Planos (contínua)

<b>Entidade: Planos</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
id	Determinante	Numérico	Inteiro 32 bits	Identificador único do plano (chave primária).
descricao	Simples	Texto	100	Descrição do plano (ex: Mensal, Trimestral, Anual).

Quadro 9 - Dicionário de dados da entidade Planos (conclusão)

Entidade: Planos				
Atributo	Classe	Domínio	Tamanho	Descrição
duracao_dias	Simples	Numérico	Inteiro 32 bits	Quantidade de dias de duração do plano.
valor	Simples	Numérico Decimal	10,2	Valor financeiro cobrado pelo plano.
ativo	Simples	Booleano	1 bit	Indica se o plano está ativo (TRUE ou FALSE).

Fonte: Elaborado pelo autor (2025)

Os Quadros 10 e 11 apresentam o DD da entidade Alunos.

Quadro 10 - Dicionário de dados da entidade Alunos (contínua)

Entidade: Alunos				
Atributo	Classe	Domínio	Tamanho	Descrição
id	Determinante	Numérico	Inteiro 32 bits	Identificador único do aluno (chave primária).
nome	Simples	Texto	100	Nome completo do aluno.
data_nascimento	Simples	Data	Date	Data de nascimento do aluno.
contato	Simples	Texto	20	Número de telefone ou contato do aluno.
email	Simples	Texto	100	Endereço de e-mail do aluno.
status	Simples	Texto (Enum)	20	Situação atual do aluno: "ativo", "inativo" ou "inadimplente".

Quadro 11 - Dicionário de dados da entidade Alunos (conclusão)

Entidade: Alunos				
Atributo	Classe	Domínio	Tamanho	Descrição
plano_id	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o plano contratado pelo aluno.
criado_por	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o usuário responsável pelo cadastro.
criado_em	Simples	Data/Hora	Timestamp	Data e hora de criação do registro.

Fonte: Elaborado pelo autor (2025)

Os Quadros 12 e 13 apresentam o DD da entidade Pagamentos.

Quadro 12 - Dicionário de dados da entidade Pagamentos (contínua)

Entidade: Pagamentos				
Atributo	Classe	Domínio	Tamanho	Descrição
id	Determinante	Numérico	Inteiro 32 bits	Identificador único do pagamento (chave primária).
aluno_id	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o aluno associado ao pagamento.
data_pagamento	Simples	Data	Date	Data em que o pagamento foi realizado.
valor	Simples	Numérico Decimal	10,2	Valor pago pelo aluno.

Quadro 14 - Dicionário de dados da entidade Pagamentos (conclusão)

<b>Entidade: Pagamentos</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
status	Simples	Texto (Enum)	20	Situação do pagamento: “pago”, “pendente” ou “atrasado”.

Fonte: Elaborado pelo autor (2025)

O Quadro 15 apresenta o DD da entidade Presencas.

Quadro 15 - Dicionário da dados da entidade Presencas

<b>Entidade: Presencas</b>				
<b>Atributo</b>	<b>Classe</b>	<b>Domínio</b>	<b>Tamanho</b>	<b>Descrição</b>
id	Determinante	Numérico	Inteiro 32 bits	Identificador único do registro de presença.
aluno_id	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o aluno presente.
instrutor_id	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o instrutor que registrou a presença.
data_presenca	Simples	Data	Date	Data em que o aluno compareceu à academia.

Fonte: Elaborado pelo autor (2025)

O Quadro 16 apresenta o DD da entidade Notificacoes.

Quadro 16 - Dicionário de dados da entidade Notificacoes

Entidade: Notificacoes				
Atributo	Classe	Domínio	Tamanho	Descrição
id	Determinante	Numérico	Inteiro 32 bits	Identificador único da notificação.
aluno_id	Composto (FK)	Numérico	Inteiro 32 bits	Chave estrangeira que referencia o aluno notificado.
mensagem	Simples	Texto	Ilimitado	Texto com o conteúdo da mensagem enviada ao aluno.
tipo	Simples	Texto (Enum)	30	Tipo da notificação: “vencimento”, “pagamento” ou “lembrete”.
data_envio	Simples	Data/Hora	Timestamp	Data e hora em que a notificação foi enviada.

Fonte: Elaborado pelo autor (2025)

## 4 DESENVOLVIMENTO DO PROJETO

O sistema desenvolvido tem como finalidade apoiar a gestão administrativa de uma academia de pequeno porte, atendendo às necessidades básicas de controle de alunos, planos, pagamentos, presenças e notificações. A aplicação foi construída utilizando o framework Laravel 12, baseado na linguagem PHP 8.4, adotando o padrão arquitetural Model–View–Controller (MVC). Essa abordagem estrutura o software em camadas independentes, facilitando a manutenção, expansão futura e organização do código-fonte.

A camada de persistência utiliza o sistema gerenciador de banco de dados PostgreSQL, escolhido por sua robustez, gratuidade e conformidade com o padrão SQL. Sua capacidade de lidar com integridade referencial e relacionamentos complexos possibilitou a implementação adequada do modelo de dados do sistema.

Para o desenvolvimento da interface, empregou-se o framework MaterializeCSS, que oferece componentes visuais responsivos compatíveis com browsers modernos. A integração entre Materialize e o mecanismo de templates Blade, nativo do Laravel, possibilitou a criação de interfaces simples, intuitivas e consistentes.

A arquitetura do sistema pode ser resumida em três camadas principais:

- **Model:** representa entidades como Aluno, Plano, Pagamento, Notificação, Presença e Usuário;
- **View:** composta pelos arquivos Blade responsáveis por exibir formulários, tabelas e dashboards;
- **Controller:** responsável por processar requisições HTTP, aplicar regras, chamar modelos e direcionar respostas às views.

### 4.1 Tecnologias Utilizadas

#### 4.1.1 PHP 8.4

A linguagem PHP foi adotada como base do projeto por ser amplamente utilizada em aplicações web, possuir grande ecossistema e apresentar compatibilidade total com o Laravel. A versão 8.4 traz melhorias importantes no desempenho, tratamento de exceções e tipagem.

#### 4.1.2 Laravel 12

O Laravel 12 foi selecionado por oferecer:

- Estrutura organizada e moderna;
- ORM Eloquent para manipulação de banco de dados de forma simplificada;
- Sistema nativo de rotas, autenticação e middleware;
- Suporte amplo da comunidade e documentação robusta.

Sua modularidade permite construir rapidamente funcionalidades essenciais, como autenticação de usuários, controle de sessão e validação de formulários.

#### 4.1.3 PostgreSQL

O PostgreSQL foi escolhido como banco de dados pela sua estabilidade, suporte a transações ACID, integridade referencial e por ser software livre. Também apresenta ótimo desempenho para buscas complexas e possui boa integração com o Laravel via Eloquent ORM.

#### 4.1.4 MaterializeCSS

Para a camada visual, foi utilizado o MaterializeCSS, um framework que segue os princípios do Material Design. A escolha foi motivada pela simplicidade de uso e pelas interfaces naturalmente responsivas, contribuindo para uma boa experiência do usuário.

### 4.2 Desenvolvimento das Funcionalidades

#### 4.2.1 Tela de Login

A tela de login é responsável pela autenticação dos usuários. O sistema utiliza o mecanismo de autenticação interno do Laravel, baseado em sessões e hash de senha via bcrypt. Após a autenticação bem-sucedida, o usuário é redirecionado ao painel correspondente ao seu perfil.

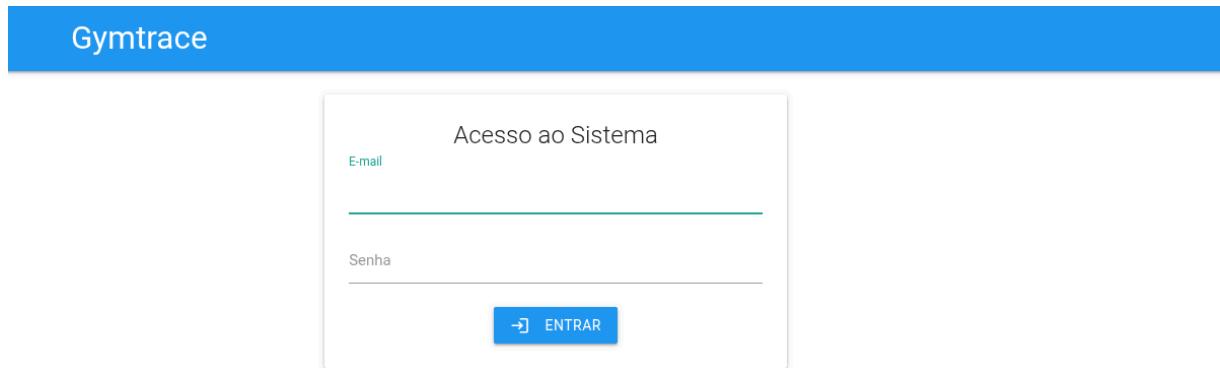


Figura 5 - Tela de Login

Fonte: Elaborado pelo autor (2025)

#### 4.2.2 Módulo do Administrador

O Administrador possui acesso completo ao sistema, sendo responsável pelo gerenciamento geral da academia. Suas funcionalidades incluem o cadastro e manutenção de alunos, planos, usuários, pagamentos e notificações, além da visualização de relatórios e indicadores no dashboard. Esse conjunto de permissões permite ao administrador controlar de forma centralizada todos os processos operacionais e administrativos, garantindo organização, segurança e eficiência na gestão da academia. Todas as funcionalidades destinadas a esse perfil são disponibilizadas por meio de telas específicas, descritas detalhadamente nas subseções seguintes.

##### 4.2.2.1 Dashboard

O *dashboard* mostra uma visão geral das informações importantes para o gestor, como a quantidade de alunos ativos, o total de planos, registros recentes e outras funções essenciais. Esta interface serve como ponto de partida para a navegação no sistema.

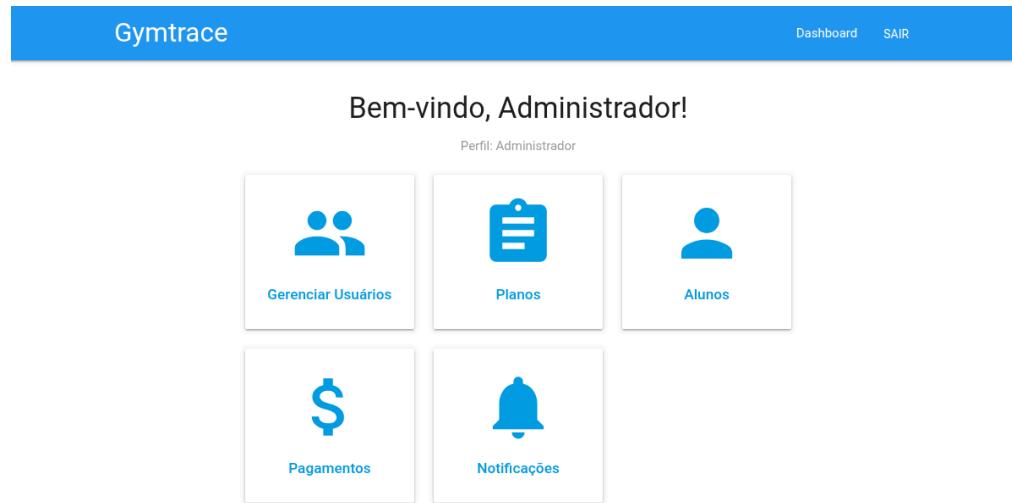


Figura 6 - Dashboard do administrador

Fonte: Elaborado pelo autor (2025)

#### 4.2.3 Módulo de Alunos

##### 4.2.3.1 Listagem de Alunos

A interface apresenta todos os alunos cadastrados, exibindo atributos como nome, situação atual e plano contratual. A listagem possui opções de visualização, edição e exclusão (somente para administradores).

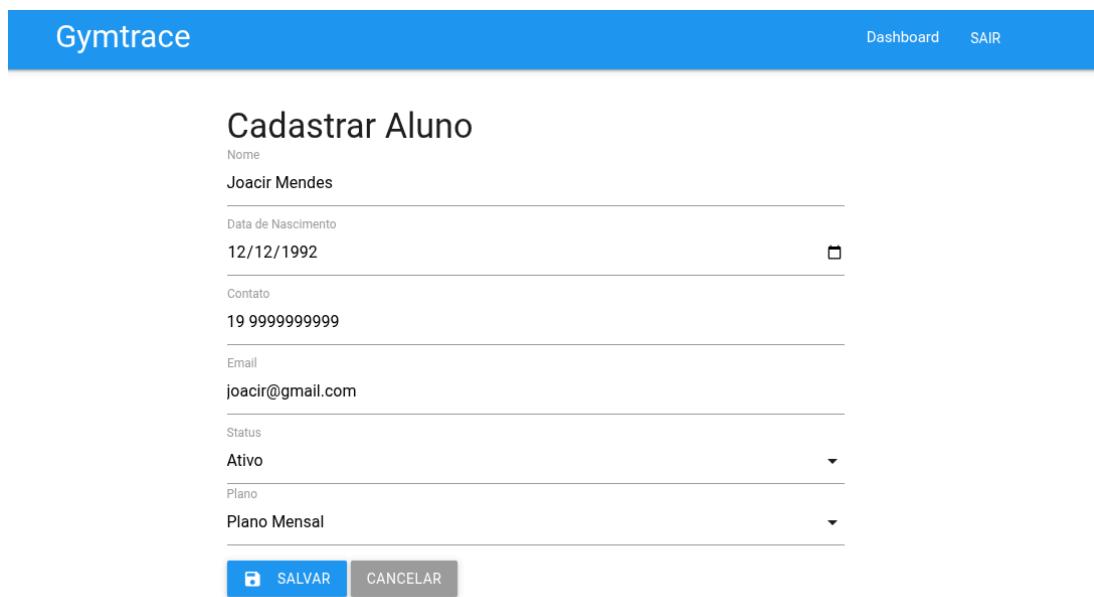
Lista de Alunos					Dashboard	SAIR
Nome	Email	Status	Piano	Ações		
Arthur Minoru Maezono	arthur@gmail.com	Ativo	Plano Mensal	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>		
João pedro	jpm83633@gmail.com	Ativo	Plano de Semestral	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>		

Figura 7 - Listagem de alunos

Fonte: Elaborado pelo autor (2025)

#### 4.2.3.2 Cadastro de Alunos

Permite adicionar dados pessoais, o plano vinculado e as informações de contato. Ao registrar um aluno, o sistema cria automaticamente um registro de pagamento com o status "pendente"



Gymtrace

Cadastrar Aluno

Nome  
Joacir Mendes

Data de Nascimento  
12/12/1992

Contato  
19 999999999

Email  
joacir@gmail.com

Status  
Ativo

Plano  
Plano Mensal

**SALVAR** CANCELAR

Figura 8 - Tela de cadastro de alunos

Fonte: Elaborado pelo autor (2025)

Nome	Email	Status	Plano	Ações
Arthur Minoru Maezono	arthur@gmail.com	Ativo	Plano Mensal	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>
Joacir Mendes	joacir@gmail.com	Ativo	Plano Mensal	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>
João pedro	jpm83633@gmail.com	Ativo	Plano de Semestral	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>

Figura 9 -Listagem de alunos após a criação de um novo aluno

Fonte: Elaborado pelo autor (2025)

#### 4.2.3.3 Edição de Alunos

Permite modificar os dados do aluno, exceto informações sensíveis que estejam relacionadas ao histórico financeiro e à presença nas aulas. Essas informações restritas não podem ser alteradas para garantir a integridade e segurança dos registros importantes.

Nome  
Joacir Mendes valor alterado

Data de Nascimento  
12/12/1992

Contato  
19 9999999999

Email  
joacir@gmail.com

Plano  
Plano de Semestral - R\$ 700,00

Status do Aluno  
Inativo

VOLTAR SALVAR ALTERAÇÕES

Figura 10 – Tela de edição de alunos

Fonte: Elaborado pelo autor (2025)

NOVO ALUNO

Aluno atualizado com sucesso!

Nome	Email	Status	Plano	Ações
Arthur Minoru Maezono	arthur@gmail.com	Ativo	Plano Mensal	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>
Joacir Mendes valor alterado	joacir@gmail.com	Inativo	Plano de Semestral	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>
João pedro	jpm83633@gmail.com	Ativo	Plano de Semestral	<a href="#">EDITAR</a> <a href="#">EXCLUIR</a>

Figura 11 – Listagem de alunos após atualização dos dados

Fonte: Elaborado pelo autor (2025)

#### 4.2.3.4 Exclusão de Alunos

A funcionalidade de exclusão possibilita ao administrador a remoção permanente de um registro de aluno do sistema. A remoção só é feita quando não afeta a integridade dos dados ligados, como pagamentos e presenças. Se houver conexões que impeçam uma exclusão segura, o sistema bloqueia a ação e informa o motivo ao usuário.

#### 4.2.4 Módulo de Planos

O módulo de planos é usado pelo administrador para criar, editar e organizar os diferentes tipos de planos que a academia oferece. Cada plano tem uma duração e um valor definidos, que depois serão ligados aos alunos.

##### 4.2.4.1 Listagem de Planos

A listagem mostra todos os planos registrados no sistema, contendo detalhes como descrição, valor, duração e situação (ativo ou inativo). A interface permite ao administrador ver rapidamente os planos disponíveis e acessar funções como editar ou excluir.

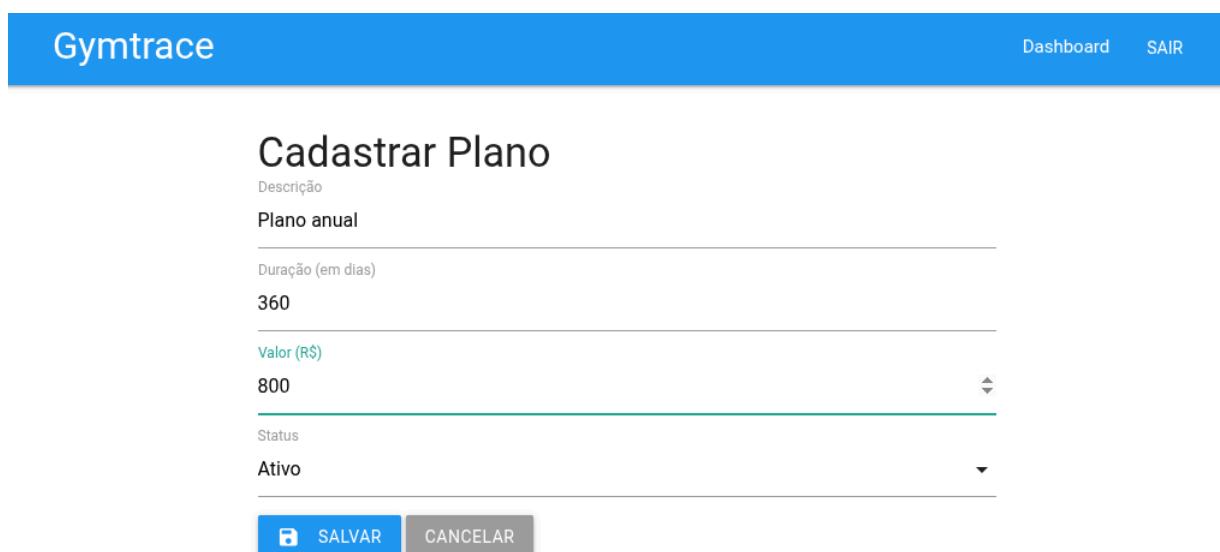
Descrição	Duração (dias)	Valor (R\$)	Ativo	Ações
Plano de Semestral	180	700,00	Ativo	
Plano Mensal	30	80,00	Ativo	

Figura 12 – Tela de Listagem de Planos

Fonte: Elaborado pelo autor (2025)

#### 4.2.4.2 Cadastro de Planos

A função de cadastro permite adicionar novos planos, definindo sua descrição, duração em dias e preço. O sistema realiza a validação dos dados para impedir a existência de campos obrigatórios não preenchidos ou contendo informações inconsistentes. Depois de cadastrado, o plano já pode ser associado aos alunos no sistema.



Gymtrace

Dashboard SAIR

### Cadastrar Plano

Descrição  
Plano anual

Duração (em dias)  
360

Valor (R\$)  
800

Status  
Ativo

■ SALVAR CANCELAR

Figura 13 – Tela de Cadastro de Planos

Fonte: Elaborado pelo autor (2025)

Descrição	Duração (dias)	Valor (R\$)	Ativo	Ações
Plano anual	360	800,00	Ativo	
Plano de Semestral	180	700,00	Ativo	
Plano Mensal	30	80,00	Ativo	

Figura 14 – Tela de Cadastro de planos após inserção do novo plano

Fonte: Elaborado pelo autor (2025)

#### 4.2.4.3 Edição de Planos

Figura 15 - Tela de Edição de Planos (Plano Anual)

Fonte: Elaborado pelo autor (2025)

Descrição	Duração (dias)	Valor (R\$)	Ativo	Ações
Plano anual (atualizado)	365	1.000,00	Ativo	
Plano de Semestral	180	700,00	Ativo	
Plano Mensal	30	80,00	Ativo	

Figura 16 – Tela de Edição de Planos: Dados Atualizados

Fonte: Elaborado pelo autor (2025)

O administrador pode alterar informações do plano conforme necessidade, como atualização de valores ou ajustes de duração. A edição preserva a integridade do sistema, pois alunos vinculados ao plano permanecem associados ao seu identificador original.

#### 4.2.4.4 Exclusão de Planos

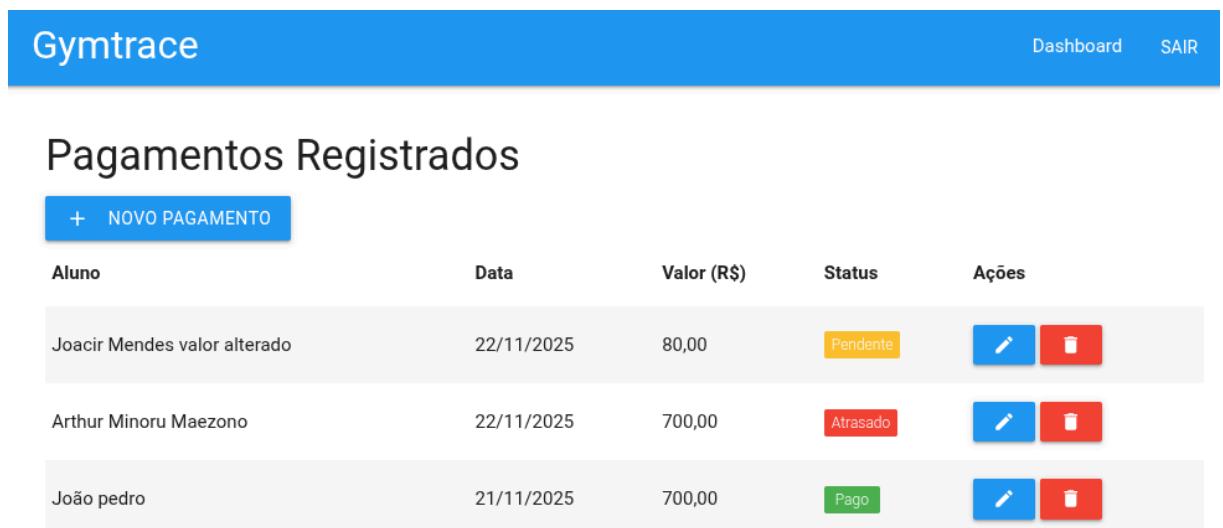
A exclusão de um plano só pode ser feita se ele não estiver ligado a nenhum aluno ativo, assegurando a consistência dos dados no sistema e prevenindo a existência de registros órfãos. Se o plano estiver associado a alunos, o sistema bloqueia a exclusão e mostra uma mensagem explicando o motivo para o usuário.

#### 4.2.5 Módulo de Pagamentos

O módulo de pagamentos permite registrar as mensalidades dos alunos, acompanhar como está a situação financeira de cada um e atualizar o status de pagamento. Esse módulo é acessado apenas pelo administrador.

#### 4.2.5.1 Listagem de Pagamentos

A listagem mostra as informações financeiras organizadas por aluno, valor, data e status do pagamento (pago, pendente ou atrasado). Essa tela ajuda o administrador a acompanhar os atrasos nos pagamentos e organizar melhor as cobranças.



Aluno	Data	Valor (R\$)	Status	Ações
Joacir Mendes valor alterado	22/11/2025	80,00	Pendente	 
Arthur Minoru Maezono	22/11/2025	700,00	Atrasado	 
João pedro	21/11/2025	700,00	Pago	 

Figura 17 – Tela de Listagem de Pagamentos

Fonte: Elaborado pelo autor (2025)

#### 4.2.5.2 Registro de Pagamentos

Ao cadastrar um aluno, o sistema gera automaticamente um pagamento com o status "pendente". No módulo de pagamentos, o administrador pode atualizar esse status para "pago", registrando a data e confirmando que a mensalidade foi paga.

Essa automação reduz a incidência de erros manuais e aprimora o controle financeiro, facilitando o acompanhamento e a organização dos pagamentos realizados pelos alunos.

Figura 18 – Tela de Cadastro/Confirmação de Pagamento

Fonte: Elaborado pelo autor (2025)

#### 4.2.5.3 Edição de Pagamentos

A edição permite corrigir datas, valores e status quando algum pagamento for registrado de forma errada. O sistema mantém um histórico correto, garantindo que os campos necessários estejam sempre validados.

Figura 19 – Tela de Edição de Pagamentos

Fonte: Elaborado pelo autor (2025)

#### 4.2.5.4 Exclusão de Pagamentos

O administrador pode apagar lançamentos incorretos ou de teste, desde que isso não afete registros de presença ou relatórios futuros. A exclusão passa por verificações internas para evitar problemas no sistema e garantir que os dados fiquem consistentes.

#### 4.2.6 Módulo de Usuários

O módulo de usuários administra os perfis dentro do sistema: Administrador e Instrutor. Cada usuário tem permissões específicas, que são controladas por meio de *middleware* para garantir o acesso correto às funções.

##### 4.2.6.1 Listagem de Usuários

A tela de listagem mostra todos os usuários registrados, exibindo nome, e-mail e tipo de perfil. O administrador tem acesso total ao módulo, podendo gerenciar todas as contas do sistema.

Nome	Email	Perfil	Ações
Administrador	admin@academia.com	Administrador	 
Instrutor 1	1@gmail.com	Instrutor	 

Figura 20 – Tela de Listagem de Usuários

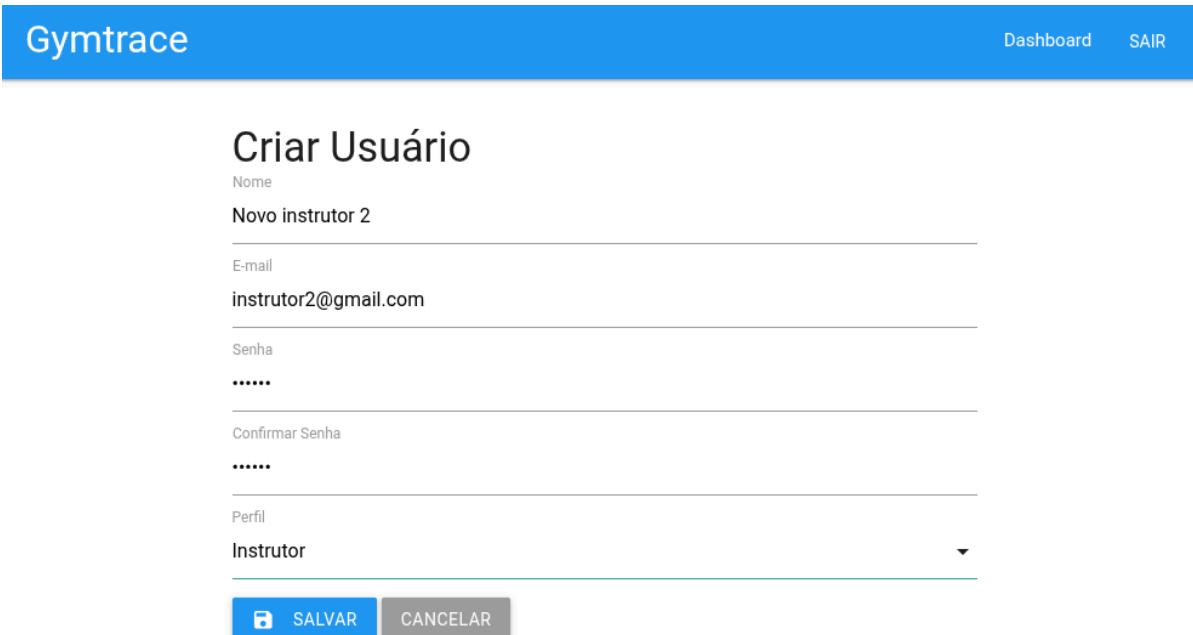
Fonte: Elaborado pelo autor (2025)

#### 4.2.6.2 Cadastro de Usuários

A criação de usuários inclui campos para nome, e-mail, senha e perfil. O sistema faz uma verificação de segurança que inclui:

- Criptografia automática da senha usando "*hash bcrypt*";
- Verificação para garantir que o e-mail seja único;
- Regras rigorosas para a senha ser forte;
- Obrigatoriedade de escolher um perfil.

Esse processo garante que o acesso às funções do sistema seja controlado de forma correta e segura.



Gymtrace

Nome

Novo instrutor 2

E-mail

instrutor2@gmail.com

Senha

.....

Confirmar Senha

.....

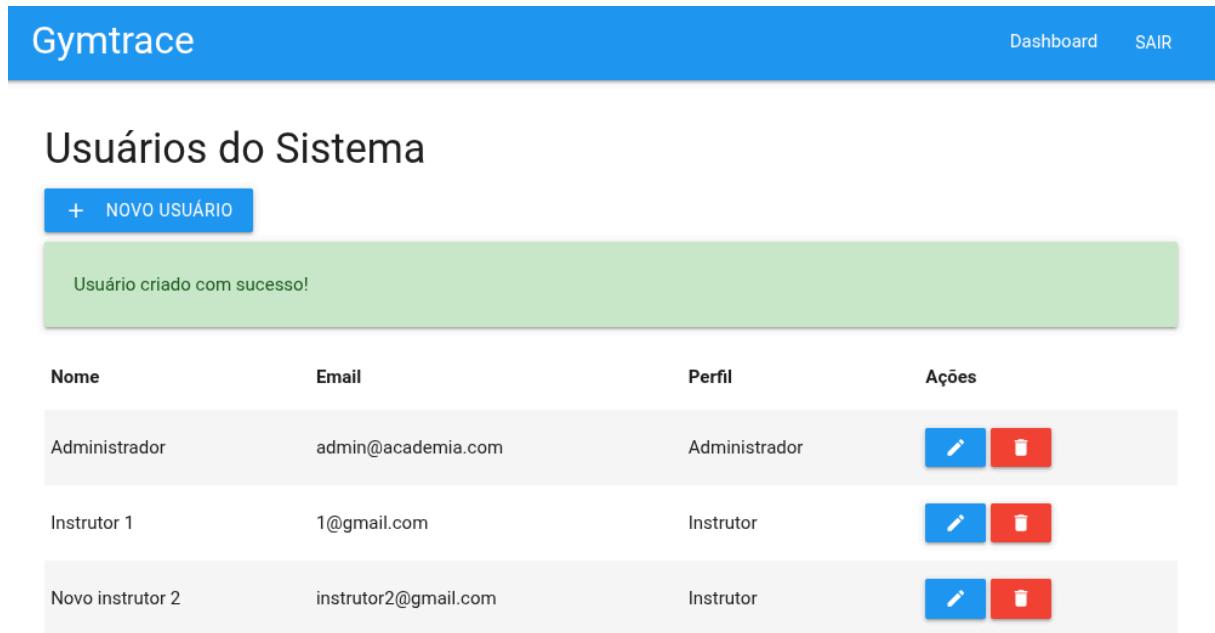
Perfil

Instrutor

**SALVAR** CANCELAR

Figura 21 – Tela de Cadastro de Usuários

Fonte: Elaborado pelo autor (2025)



Nome	Email	Perfil	Ações
Administrador	admin@academia.com	Administrador	 
Instrutor 1	1@gmail.com	Instrutor	 
Novo instrutor 2	instrutor2@gmail.com	Instrutor	 

Figura 22 – Tela de Listagem de Usuários: Novo Usuário Cadastrado

Fonte: Elaborado pelo autor (2025)

#### 4.2.6.3 Edição de Usuários

A edição permite alterar informações como nome, e-mail e perfil. A mudança de senha é opcional e, se feita, a senha é protegida automaticamente por criptografia. Isso garante segurança e impede que dados sensíveis fiquem expostos.

Gymtrace

Dashboard SAIR

## Editar Usuário

Nome  
Novo instrutor 2.0 (atualizado)

E-mail  
instructor2Atualizado@gmail.com

Nova Senha (opcional)

Confirmar Senha

Perfil  
Instrutor

**ATUALIZAR** CANCELAR

Figura 23 – Tela de Edição de Usuários

Fonte: Elaborado pelo autor (2025)

Gymtrace

Dashboard SAIR

## Usuários do Sistema

+ NOVO USUÁRIO

Usuário atualizado com sucesso!

Nome	Email	Perfil	Ações
Administrador	admin@academia.com	Administrador	
Instrutor 1	1@gmail.com	Instrutor	
Novo instrutor 2.0 (atualizado)	instructor2Atualizado@gmail.com	Instrutor	

Figura 24 – Tela de Edição de Usuários: Perfil Atualizado

Fonte: Elaborado pelo autor (2025)

#### 4.2.6.4 Exclusão de Usuários

A exclusão apaga o usuário do sistema para sempre. O administrador pode excluir somente usuários que não estejam ativos, mantendo a organização do banco de dados e a segurança. O sistema não permite que o próprio administrador que está conectado se exclua, evitando que ele perca o acesso ao sistema.

#### 4.2.7 Módulo de Notificações

O módulo de notificações permite ao administrador ver e gerenciar mensagens enviadas para os alunos, especialmente avisos sobre pagamentos em atraso.

##### 4.2.7.1 Listagem de Notificações

O sistema mostra todas as notificações registradas, incluindo o aluno que recebeu, a data e a descrição da mensagem. Essa função ajuda a controlar o histórico das comunicações feitas no sistema.

Aluno	Tipo	Mensagem	Data de Envio	Ações
João pedro	Vencimento	Prezado aluno seu plano venceu blablabla...	21/11/2025 16:55	

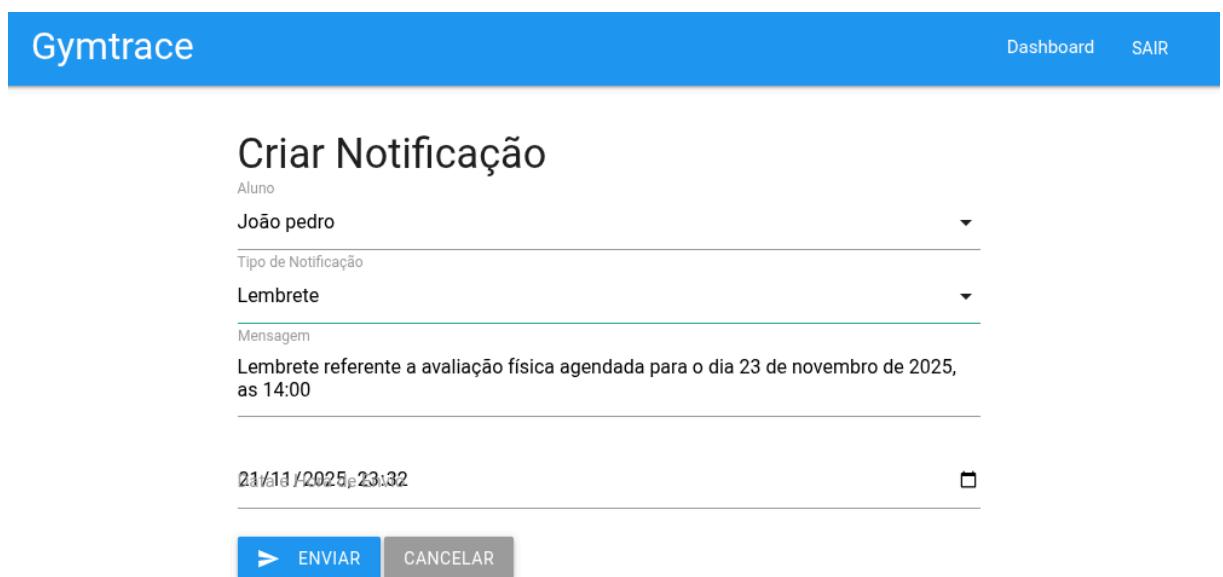
Figura 25 – Listagem de Notificações

Fonte: Elaborado pelo autor (2025)

##### 4.2.7.1 Criação de Notificações

O administrador pode criar novas notificações manualmente, que são salvas no banco de dados e usadas em campanhas internas ou alertas. Além disso, o

sistema envia e-mails automaticamente em caso de atraso no pagamento. Essa função usa o componente Mail do Laravel junto com um servidor SMTP do Gmail. O envio acontece sempre que o sistema identifica um pagamento pendente que passou do prazo de vencimento.



Gymtrace

Dashboard SAIR

## Criar Notificação

Aluno

João pedro

Tipo de Notificação

Lembrete

Mensagem

Lembrete referente a avaliação física agendada para o dia 23 de novembro de 2025, as 14:00

21/11/2025 23:32

ENVIAR CANCELAR

Figura 26 – Tela de Criação de Notificação

Fonte: Elaborado pelo autor (2025)

### 4.3 Módulos destinados ao Instrutor

O Instrutor tem acesso limitado no sistema, conforme as permissões definidas. Seu trabalho é consultar e acompanhar os alunos e registrar as presenças diárias, mas ele não pode criar, alterar ou apagar cadastros administrativos. Essa configuração garante a segurança e mantém os dados protegidos, ao mesmo tempo que oferece ao instrutor as ferramentas necessárias para acompanhar os treinos.

#### 4.3.1 Dashboard (Instrutor)

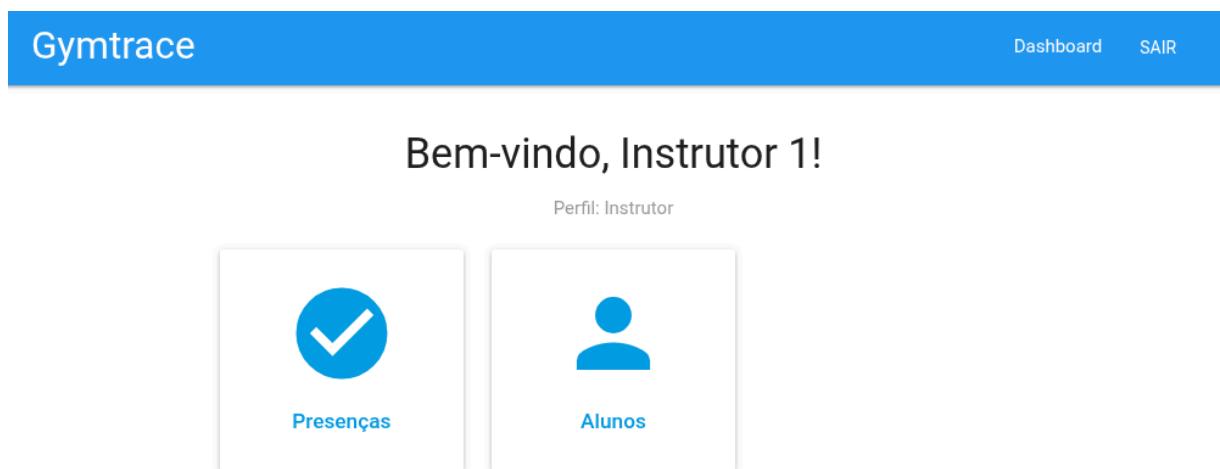


Figura 27 – Tela de dashboard do instrutor

Fonte: Elaborado pelo autor (2025)

#### 4.3.2 Consulta de Alunos (Instrutor)

O Instrutor tem acesso a uma lista completa dos alunos cadastrados, podendo ver informações básicas, detalhes de cada aluno e o plano que ele está usando no momento. Essa função ajuda o instrutor a acompanhar diariamente os treinos e a presença dos alunos.

#### 4.3.3 Listagem de Alunos (Somente consulta)

A lista mostra informações importantes, como nome, contato e situação do aluno. Diferente do Administrador, o Instrutor não pode cadastrar, editar ou excluir dados, ele vê apenas uma versão simples da tela para visualização.

Nome	Email	Status	Plano	Ações
Arthur Minoru Maezono	arthur@gmail.com	Ativo	Plano Mensal	
Joacir Mendes valor alterado	joacir@gmail.com	Inativo	Plano de Semestral	
João pedro	jpm83633@gmail.com	Ativo	Plano de Semestral	

Figura 28 – Listagem de Alunos para Instrutor

Fonte: Elaborado pelo autor (2025)

#### 4.3.4 Registro de Presenças

O módulo de presenças é a principal função do instrutor. Ele permite registrar a entrada dos alunos na academia, criando um histórico de presença que pode ser consultado depois pela administração.

#### 4.3.5 Listagem de Presenças

A tela de listagem mostra todos os registros de presença que o instrutor marcou, incluindo:

- Nome do aluno
- Data da presença
- Instrutor responsável

Essa lista serve tanto como histórico quanto como um relatório que ajuda a controlar a frequência dos alunos na academia.

Gymtrace		Dashboard	SAIR
Registro de Presenças			
<a href="#">+ NOVA PRESENÇA</a>			
Aluno	Instrutor	Data	Ações
João pedro	Instrutor 1	21/11/2025	<a href="#"></a> <a href="#"></a>

Figura 29 – Listagem de Presenças

Fonte: Elaborado pelo autor (2025)

#### 4.3.6 Registro de Nova Presença

O instrutor pode registrar uma nova presença escolhendo:

- O aluno que está presente;
- A data (geralmente já preenchida automaticamente).

Essa função garante o controle diário dos alunos que frequentam a academia.

O registro é feito por meio de um formulário simples, onde o instrutor seleciona o aluno em um menu suspenso. Depois de confirmado, o sistema cria automaticamente o registro e o vincula ao próprio instrutor.

Gymtrace		Dashboard	SAIR
Registrar Presença			
<input type="text" value="Aluno"/>			
Arthur Minoru Maezono			
<input type="text" value="Instrutor"/>			
Novo instrutor 2.0 (atualizado)			
<input type="text" value="Data da Presença"/>			
21/11/2025			
<a href="#"> SALVAR</a>		CANCELAR	

Figura 30 – Tela de Registro de Presença

Fonte: Elaborado pelo autor (2025)

#### 4.3.7 Limitações de Acesso do Instrutor

O Instrutor tem acesso somente às funções de consulta aos alunos e registro das presenças. As outras funções (como cadastros, planos, pagamentos, usuários e notificações) ficam escondidas ou bloqueadas pelo sistema de permissões do *middleware*.

Essa configuração assegura que apenas o administrador tenha privilégios para executar operações críticas, mantendo a segurança e preservando a integridade do sistema.

### 4.4 Considerações sobre Segurança

A segurança do sistema inclui:

- *Hash* de senha (bcrypt), que protege as senhas ao transformar em códigos difíceis de quebrar;
- *Middleware* de autenticação, que controla o acesso permitindo apenas usuários logados;
- Sistema de permissões, onde administradores e instrutores têm diferentes níveis de acesso;
- Validação dos formulários, que evita o envio de dados errados;
- Proteção de rotas sensíveis, usando *middleware* personalizado para maior segurança.

Essas medidas garantem que as informações estejam protegidas e impedem acessos não autorizados, mantendo a integridade do sistema.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho enfrentou o desafio central identificado na introdução: a dependência de métodos manuais por academias de pequeno porte, que gera erros operacionais e inadimplência elevada (20-30% dos mensalistas), agravada pela inviabilidade financeira de sistemas comerciais complexos e custosos.

### 5.1 Resultados Alcançados

O desenvolvimento do Gymtrace entregou uma solução técnica completa e documentada que responde diretamente a essa problemática. Especificamente, o projeto alcançou:

- Especificação de um MVP funcional com módulos essenciais: cadastro de alunos, gestão de planos, controle de pagamentos, registro de presença e notificações;
- Modelagem de dados robusta, representada através de Diagrama de Classes, DER e Dicionário de Dados, garantindo uma base sólida para a persistência de informações;
- Arquitetura tecnológica definida com Laravel e PostgreSQL, assegurando segurança, desempenho e custo zero de licenciamento;
- Documentação técnica completa que permite a implementação prática do sistema.

### 5.2 Impacto esperado nas Academias

A adoção do Gymtrace tem o potencial de gerar transformações significativas na gestão de academias de bairro:

- Redução de erros operacionais pela automação de processos antes manuais (planilhas, cadernos);
- Diminuição da inadimplência através do controle rigoroso de pagamentos e notificações automáticas;
- Maior previsibilidade financeira com relatórios gerenciais básicos sobre receita e inadimplência;

- Otimização de tempo para gestores e instrutores, que podem focar em atividades de maior valor.

### **5.3 Limitações do Estudo**

Reconhecem-se as seguintes limitações inerentes ao escopo deste trabalho:

- Restrição funcional a módulos administrativos básicos, excluindo controle detalhado de treinos e integrações com hardware;
- Foco em MVP sem a implementação de funcionalidades avançadas como inteligência de negócios (BI) ou automação de marketing;
- Validação em ambiente controlado, necessitando de testes em cenários reais de operação para ajustes finos;
- Geração de relatórios básicos, com potencial expansão para análises mais sofisticadas de churn e performance.

### **5.4 Propostas de desenvolvimento e melhorias**

Para a evolução do Gymtrace, propõe-se a seguinte ordem prioritária de desenvolvimentos futuros:

1. Implementação e Testes em Ambiente Real – Validação do MVP com academias parceiras para coleta de feedback e identificação de melhorias críticas.
2. Desenvolvimento de Módulo de Treinos – Sistema para criação e acompanhamento de fichas de exercícios personalizadas, integrado ao cadastro de alunos.
3. Integração com Gateways de Pagamento – Implementação de soluções como Pix, Mercado Pago ou Stripe para automatizar cobranças e reduzir ainda mais a inadimplência.
4. Desenvolvimento do Aplicativo Móvel – Versão mobile para alunos consultarem horários, planos e receberem notificações, utilizando React Native ou Flutter.
5. Expansão para Módulo Comercial – Funcionalidades para vendas de suplementos e produtos, integrando gestão de estoque e e-commerce.

6. Integração com Hardware – Conectores para catracas eletrônicas e sensores para automatizar registro de presença e acesso.

A concretização desta roadmap transformaria o Gymtrace de um sistema de gestão básico para uma plataforma integral de administração para academias de pequeno porte, sempre mantendo os princípios de simplicidade, baixo custo e código aberto que nortearam este trabalho.

## 5.5 Acesso aos repositórios

O projeto conta com um QR Code que fornece acesso direto ao seu repositório no GitHub. A leitura do código permite a visualização do código-fonte, da documentação técnica e do histórico de desenvolvimento da aplicação. O repositório mantém aberto todo o material relacionado à implementação e evolução do trabalho.

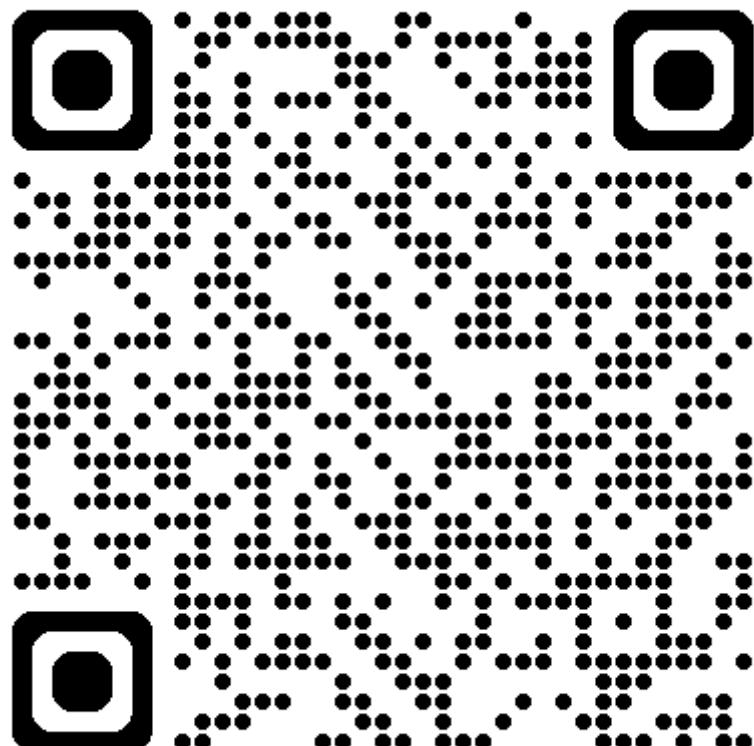


Figura 31 - Acesso ao repositório do projeto

Fonte: Elaborado pelo autor (2025)

## REFERÊNCIAS BIBLIOGRÁFICAS

- ACAD BRASIL. **Relatório de benchmarking do setor de fitness de 2025.** [S. I.]: ACAD Brasil, 2025. Disponível em: <https://acadbrasil.com.br/pagina-informativo/hfa-publica-relatorio-de-benchmarking-do-setor-de-fitness-de-2025/>. Acesso em: 23 nov. 2025.
- CONSELHO FEDERAL DE EDUCAÇÃO FÍSICA (CONFEF). **25 anos de profissão regulamentada.** Rio de Janeiro, n. 83, dez. 2023. Disponível em: <https://www.confef.org.br/>. Acesso em: 23 nov. 2025.
- DATE, C. J. **Database Design and Relational Theory: Normal Forms and All That Jazz.** 2. ed. Healdsburg, California: Apress, 2019.
- ELITE VALE. **Setor fitness se expande e número de academias quase triplica no Brasil em dez anos.** [S. I.], 2025. Disponível em: <https://www.elitevale.com.br/materias/setor-fitness-se-expande-e-numero-de-academias-quase-triplica-no-brasil-em-dez-anos/>. Publicado em: 13 out. 2025. Acesso em: 23 nov. 2025.
- FITNESS BRASIL. **Panorama Setorial Fitness Brasil 2024 – 3ª edição.** [S. I.]: Fitness Brasil, 2024. Disponível em: <https://www.fitnessbrasil.com.br/newsfitbr/descubra-as-tendencias-do-mercado-fitness-em-2024-uma-analise-completa-do-panorama-setorial-fb/>. Acesso em: 23 nov. 2025.
- HUBFIT. **Sistema de Gestão para Academias - Hubfit.** Balneário Camboriú, 2024. Disponível em: <https://hubfit.com.br/>. Acesso em: 23 nov. 2025.
- INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). **Censo da educação superior até 2022.** Disponível em: <https://www.gov.br/inep/pt-br>. Acesso em: 21 out. 2025.
- LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais.** 11. ed. São Paulo: Pearson Education do Brasil, 2014.
- LARAVEL. **Laravel Documentation.** [S. I.], 2024. Disponível em: <https://laravel.com/docs>. Acesso em: 23 nov. 2025.
- LARAVEL. **The PHP Framework For Web Artisans.** [S. I.], 2024. Disponível em: <https://laravel.com/>. Acesso em: 23 nov. 2025.
- LOPES, Alexandrino Tiago Faustino; DANTAS, Henrique Ferreira. **Entre halteres e hardwares: a revolução digital nas academias de pequeno porte.** Orientador: Prof. Me. Alberto Bruno de Oliveira Viana. 2024. 34 f. Trabalho de Conclusão de Curso (Tecnólogo em Sistemas para Internet) - Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – Campus Porto Velho Zona Norte, Porto Velho, 2024. Disponível em: <https://repositorio.ifro.edu.br/items/73881565-06b4-4f5e-bd83-81c0ba730fe3>. Acesso em: 23 nov. 2025.

LOPES DE ALMEIDA, A.; FROHLICH DA SILVA, R. **Desenvolvimento de um sistema para academias com acompanhamento da rotina de treino.** Santa Maria: Universidade Franciscana, 2020. Disponível em: <[https://tfgonline.lapinf.ufn.edu.br/media/midias/Ariel\\_Lopes\\_de\\_Almeida\\_17fn6uR.pdf](https://tfgonline.lapinf.ufn.edu.br/media/midias/Ariel_Lopes_de_Almeida_17fn6uR.pdf)>. Acesso em: 23 nov. 2025.

NEOTECHY. ***In-Depth Comparison of Laravel, Symfony, and CodeIgniter.*** [S. I.], 17 dez. 2024. Disponível em: <https://dev.to/neotechy/exploring-php-frameworks-in-depth-comparison-of-laravel-symfony-and-codeigniter-2453>. Acesso em: 23 nov. 2025.

NEXT FIT. **Sistema para Academia - Next Fit.** [S. I.], 2024. Disponível em: <https://www.nextfit.com.br/>. Acesso em: 23 nov. 2025.

OBJECT MANAGEMENT GROUP (OMG). ***Unified Modeling Language: Infrastructure, version 2.5.*** [S. I.]: Object Management Group, 2015. Disponível em: <https://www.omg.org/spec/UML/2.5>. Acesso em: 23 nov. 2025.

PHP MANUAL. ***PHP Manual.*** [S. I.]: [s. d.]. Disponível em: [https://www.php.net/manual/pt\\_BR/](https://www.php.net/manual/pt_BR/). Acesso em: 23 nov. 2025.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: Uma Abordagem Profissional.** 8. ed. Porto Alegre: AMGH, 2016.

POSTGRESQL. ***PostgreSQL: The World 's Most Advanced Open Source Relational Database.*** [S. I.], 2024. Disponível em: <https://www.postgresql.org/>. Acesso em: 23 nov. 2025.

RIES, Eric. **A startup enxuta.** São Paulo: Lua de Papel, 2012. 209 p. ISBN 9788581780139.

SEBRAE-PR. **Mercado de academias no Brasil: dados e tendências.** Disponível em: <https://sebraepr.com.br/impulsiona/mercado-de-academias-no-brasil-dados-e-tendencias/>. Curitiba, 15 dez. 2024. Acesso em: 23 nov. 2025.

SCHWABER, K.; BEEDLE, M. ***Agile Software Development with Scrum.*** Upper Saddle River, NJ: Prentice Hall, 2002.

SCHWABER, K. ***Scrum Guide.*** [S. I.], Nov. 2020. Disponível em: <https://www.scrumguides.org/>. Acesso em: 23 nov. 2025.

SOMMERVILLE, I. **Engenharia de Software.** 10. ed. São Paulo: Pearson Education do Brasil, 2016.

STACK OVERFLOW. ***Developer Survey Results 2024.*** Disponível em: <https://insights.stackoverflow.com/survey/2024>. Acesso em: 22 out. 2025.

STAIR, R.; REYNOLDS, G. **Princípios de Sistemas de Informação**. Tradução da 11. ed. norte-americana. São Paulo: Cengage Learning, 2015.

TECNOFIT. **Sistema de Gestão para Academias - Tecnofit**. Curitiba, 2024. Disponível em: <https://tecnofit.com.br/>. 23 nov. 2025.

TERRA. **Número de academias no Brasil quase triplica em 10 anos**. Disponível em: <https://www.terra.com.br/noticias/numero-de-academias-no-brasil-quase-triplica-em-10-anos,2434410e4255e97893c6fc219e07f12ey9os28sa.html>. [S. I.], 2025. Publicado em: 16 mar. 2025. Acesso em: 23 nov. 2025.